T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DOCTORAL THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

Gökhan ERCAN

SEMANTIC RELATION EXTRACTION BY ENRICHING WORD EMBEDDINGS EXPLOITING TURKISH MORPHOLOGY

SUPERVISOR
Prof. Dr. Olcay Taner YILDIZ

İSTANBUL, March 2025

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DOCTORAL THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

Gökhan ERCAN

(214DCS8012)

SEMANTIC RELATION EXTRACTION BY ENRICHING WORD EMBEDDINGS EXPLOITING TURKISH MORPHOLOGY

SUPERVISOR

Prof. Dr. Olcay Taner YILDIZ

İSTANBUL, March 2025

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DOCTORAL THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

Gökhan ERCAN

(214DCS8012)

SEMANTIC RELATION EXTRACTION BY ENRICHING WORD EMBEDDINGS EXPLOITING TURKISH MORPHOLOGY

Date:

Thesis Supervisor: Prof. Dr. Olcay Taner YILDIZ / Özyeğin University

Jury Members: Assoc. Prof. Arzucan ÖZGÜR / Boğaziçi University

Asst. Prof. Zeynep İlknur KARADENİZ / Özyeğin University

Asst. Prof. Emine EKİN / Işık University

Asst. Prof. Ahmet Feyzi ATEŞ / Işık University

İSTANBUL, March 2025

ÖZET

ANLAMSAL İLİŞKİ ÇIKARIMINDA TÜRKÇE MORFOLOJİSİ KULLANILARAK DAĞITIK KELİME GÖSTERİMLERİNİN ZENGİNLEŞTİRİLMESİ

Dağıtık kelime gösterimleri (DG), metinsel veri içindeki kelime dağılım ilişkilerinin analiz edilmesiyle dildeki anlamsal ve sözdizimsel düzenlerin yakalanması için kullanılır. DG üreten modelleme yöntemleri, dilin doğasından gelen *"aynı bağlam içerisinde yer alan kelimeler, birbirlerine yakın anlamlara sahip olma eğilimi gösterir"* varsayımına (dağılımsal hipotez) dayanmaktadır. Bu modelleme yöntemleri, gözetimsiz doğaları sayesinde insan yargı girdisi olmaksızın eğitilebilmekte, bu da araştırmacıların görece düşük maliyetlerle büyük veri kümelerini eğitebilmelerine olanak sağlamaktadır. Kelime-bazlı modeller İngilizce gibi sınırlı dağarcığa sahip dillerde iyi çalışmakla birlikte Türkçe gibi morfolojik açıdan zengin, sınırsız dağarcığa sahip dillerde oldukça verimsizdir. Dağarcık-dışı-kelimeler ve az-geçen-kelimeler problemlerine çözüm sunan kelime-altı modellemede yaygın olarak kullanılan n-gram ve istatistiksel ayrıştırma yöntemlerinin ortografik benzerliğe karşı hassas olduğu, dolayısıyla ilişkisiz kavramları (*enişte* - *erişte*) birbirinden ayıramadığını tespit ettik. Morfolojik ayrıştırma yönteminin ise bu tür problemlere etkisinin literatürde tutarsız sonuçlar gösterdiği saptanmıştır.

Bu tez farklı anlam ilişkisi türleri (ilişkisellik ve benzerlik vb.) üzerine kavramsal varsayım ve geliştirmeler yapmayı, dil morfolojisini girdi olarak modellemenin kelime-altı DG modelleri üzerindeki rolünü ve bu etkiyi ölçebilmek için gerekli olan veri kümesi üretme metodolojilerini ve değerlendirme yöntemlerini geliştirmeyi amaçlamaktadır. Çalışma kapsamında farklı model ve ayrıştırma yöntemleri ampirik olarak denenmiş, AnlamVer ve OSimUnr kelime çifti veri kümeleri üretilmiş, ve ayrıştırmanın modele eklediği gürültüyü ölçebilmek için *ilişkisellik sınıflandırma* görevi ve ilgili ölçme

yöntemleri önerilmiştir. Deneylerimiz, morfolojik ayrıştırmanın n-gram bazlı yöntemlere oranla çok daha az gürültü ürettiğini ve görevin doğasına bağlı olarak ciddi bir performans artışı sağlayabileceğini göstermektedir.

Anahtar Kelimeler: Türkçe Morfolojisi, Kelime Gösterimleri, Anlamsal Model, Anlamsal İlişkisellik, Ortografik Benzerlik

ABSTRACT

SEMANTIC RELATION EXTRACTION BY ENRICHING WORD EMBEDDINGS EXPLOITING TURKISH MORPHOLOGY

Distributed representations (DR) are used to capture semantic and syntactic patterns in language by analyzing the distributional relationships of words within textual data. The modeling methods that produce DR are based on the assumption (distributional hypothesis) that *"words that occur in the same context tend to have similar meanings*," which is inherent to the nature of language. These modeling methods, due to their unsupervised nature, can be trained without human judgment input, allowing researchers to train large datasets at relatively low costs. Although word-based models perform effectively for languages with limited vocabularies, such as English, they exhibit considerable inefficiency when applied to morphologically rich languages with unlimited vocabularies, such as Turkish. We observed that n-gram and statistical segmentation methods, which are commonly used in subword modeling to address the issues of out-of-vocabulary and rare-words, are highly sensitive to orthographic similarity. Consequently, these methods struggle to distinguish between unrelated concepts (e.g., *shrink* - *shrine*). Moreover, we noted that the impact of morphological segmentation methods on these types of problems has shown inconsistent results in the literature.

This thesis aims to make conceptual assumptions and improvements concerning different types of semantic relationships (e.g., relatedness and similarity), to model the role of language morphology as an input in subword DR models, and to develop the dataset generation methodologies and evaluation methods to measure this effect. Within the scope of the study, different models and segmentation methods were empirically tested, the AnlamVer and OSimUnr datasets were produced, and the task of *relatedness classification* and associated

evaluation methods were proposed to measure the noise introduced by segmentation to the model. Our experiments demonstrate that morphological segmentation produces significantly less noise compared to n-gram-based methods and can lead to substantial performance improvements depending on the nature of the task.

Keywords: Turkish Morphology, Word Embeddings, Semantic Model, Semantic Relatedness, Orthographic Similarity

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, Prof. Dr. Olcay Taner Yıldız, for his expertise, support, guidance, and friendship throughout the years. I also thank the many linguists, researchers, and annotators in his lab whose contributions made the language resources used in this study possible.

A special thanks to my friend and mentor Kumsal Obuz for his valuable contributions as a second eye, and for enduring my long and loud presentations with kindness.

Lastly, but most importantly, to my family — Canan Ercan, Gökçe Ercan Mildon, Can Mildon, and my lovely wife Dina — thank you for your endless love, encouragement, and support throughout this long and terrible life decision.

Gökhan ERCAN

In loving memory of my father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS LIST

acc: Accuracy

B: Billion

BPE: Byte Pair Encoding

CB: CBOW - Continuous Bag of Words - default objective

CDS: Context Distribution Smoothing

CG: Char-Gram

CL: Computational Linguistics

DR: Distributed Representation

DS: Distributional Semantics

DSM: Distributional Semantic Model

editsim: Orthographic similarity score calculated with inverted version of normalized edit distance

EN (en): English language

err: Error

FN: False Negative

FP: False Positive

FST: Finite State Transducer

FT: Subword-level FastText model and implementation

FT-CG: FastText model with the default Char-Gram[3–6] segmentation

FT-HYP: FastText model with Hyphenation or syllables segmentation

FT-M: FastText model with full Morphology

FT-MR: FastText model with morphological Roots only

GB: Gigabyte

GPU: Graphics Processing Unit

HREL: Highly Related

HYP: Hyphenation

IC: Information Content

IR: Information Retrieval

LCS: Longest Common Subsequence

LLM: Large Language Model

LM: Language Model

M: Million

MR: Morphological Roots

NER: Named Entity Recognition

NLTK: Natural Language Toolkit

NLP: Natural Language Processing

NLU: Natural Language Understanding

ODIS: Orthographically Dissimilar

OOV: Out of Vocabulary

OSim: Orthographic Similarity (i.e., string/textual similarity) score of words calculated by editsim or over_ft23

OSimUnr: Orthographically Similar but Semantically Unrelated

over_ft23: Normalized textual similarity metric calculating the overlapping factor of FastText n-grams in [2–3] Char-Gram setting

PMI: Pointwise Mutual Information

POS: Part of Speech

PPMI: Positive Pointwise Mutual Information

Q3: Word-pairs where OSim scores are between 0.5 and 0.75

Q4: Word-pairs where OSim scores are between 0.75 and 1

REL: Related

rel: Relatedness

SCI: Semantic Clarity Index

SG: SkipGram objective - non-default objective

sim: Similarity

SU: Similar Unrelated

SVD: Singular Value Decomposition

TF/IDF: Term Frequency / Inverse Document Frequency

TR (tr): Turkish language

TN: True Negative

TP: True Positive

UNR: Unrelated

VSM: Vector Space Model

W2V: Word-level Word2Vec model and implementation

CHAPTER 1

1. INTRODUCTION

It has always been a challenging but rewarding task for researchers to make computers understand natural languages. Although computers are highly competent at interpreting formal languages, they are not capable of understanding natural languages at all. The primary reason for this challenge is that language itself is a social institute which is constantly evolving with humankind, rather than being an individual function of a speaker or writer (De Saussure et al., 2011). Language is subjective by its nature. As one of the founders of modern linguistics, Ferdinand de Saussure, stated: "speaking of linguistic law in general is like trying to pin down a ghost" (De Saussure et al., 2011). Saussure also noted that *writing* is one of the *signs* of a language, and two should not be confused with each other. Humankind has developed writing with its grammatical rules to make information permanent. One might consider the syntactic and semantic structures of a language as constant phenomena at first glance, but it is inevitable that both will change as language and culture evolve over time.

Another challenge in enabling computers to understand languages lies in representing language on a computer. Since language, as a social institution, cannot be directly encoded on a computer, natural language processing (NLP) researchers have had no choice but to use *writing* and its alphabet as a representation of language. Although letters and words can be easily represented digitally, this simplified model may lose the social or individual intent of the message for both the sender and the receiver. The understanding and learning capabilities of computer systems are inherently limited by their ability to represent information.

Over the last three decades, numerous highly successful NLP methods and applications have been developed, with some becoming integral to our everyday

lives. Applications such as next word prediction, spelling correction, question answering, summarization, and machine translation systems are just a few examples of the valuable contributions of NLP. With the advance of deep learning methods (LeCun et al., 2015) in the last decade, these applications have seen significant improvements in accuracy, context awareness, and overall performance, further enhancing their impact and utility in everyday tasks. As the volume of publicly available digital documents continues to grow in today's digital age, unsupervised machine learning methods have gained significant popularity within the NLP community. These methods allow researchers to train models on vast amounts of data, potentially leading to more efficient models with better generalization capabilities. Unlabeled datasets can be collected at a fraction of the cost of labeled ones, as they do not require human judgment for each item. While unsupervised representation learning—often referred to as **self-supervised** learning, popularized by Yann LeCun (LeCun and Misra, 2021)—performs well on their own, they are also increasingly used as a cost-effective pre-training stage for supervised learning tasks, offering additional efficiency (Turian et al., 2010). This approach has been further enhanced by the development of foundational models and large language models (LLMs), which leverage vast amounts of unlabeled data to create highly versatile representations (Zhao et al., 2023). These models not only improve the performance of specific NLP tasks but also serve as a robust foundation for a wide range of downstream applications.

1.1 MOTIVATION

To understand a book, one must first understand its paragraphs; to understand a paragraph, one must grasp its sentences. Finally, to comprehend a sentence, the meaning of each word must be understood. There are numerous definitions of the term *meaning* in linguistics. However, NLP, computational linguistics (CL), and information retrieval researchers often consider meaning in a narrower, statistical sense, as advocated by Peter Norvig in the famous debate,

Two Cultures of Statistical Learning (Norvig, 2011). The main intuition behind distributional semantics (DS) research is that the co-occurrence of linguistic elements (usually words) within text can provide insights into their meaning. This intuition is quite old, as DS research, commonly referred to Harris's so-called distributional hypothesis, states: "words that occur in similar contexts tend to have similar meanings" (Harris, 1954).

Under the assumption of the distributional hypothesis, distributional semantic models (DSMs) aim to learn best possible vector representations (i.e., embeddings) of linguistic items, with the expectation that these representations capture meaningful relations from text. Due to their unsupervised nature, these modeling techniques do not require any human judgement input to train, which allows researchers to train very large datasets in relatively low costs. However, the evaluation of such models is still subject to human judgement as a "gold standard", typically gathered through questions like "How two words evidence and proof similar on a scale from 0 to 10" or "What is X to *Turkey* as *Paris* to *France*".

The earliest studies on vector semantics research date back to 1992 (Schutze, 1992; Schütze and Pedersen, 1993). Most of these methods are count-based, utilizing singular value decomposition (SVD) and similar sparse matrix approaches, which have never been scalable in terms of memory usage or computational complexity (Baroni et al., 2014). Since 2013, a study by Mikolov et al. (2013a) has gained increasing attention by demonstrating that prediction-based models (dense vectors trained directly through neural networks) are scalable enough to train on 100 billion-word datasets in a single day, even on a single machine implementation. Additionally, Mikolov et al. (2013d) showed that performing basic algebraic operations on learned vectors could reveal many meaningful semantic and syntactic regularities, a phenomenon referred to as *compositionality*. A well-known example of compositionality is the operation King - Man + Woman, which yields a vector very close to Queen (Mikolov et al., 2013d). This example illustrates that the '*female – male* relationship' can be

automatically encoded in word embeddings, serving as evidence of the generalization power of such representations.

DS research has traditionally used words as the smallest meaningful linguistic unit of a language. This assumption works well for languages with limited vocabulary and minimal inflection, such as English. In addition to English being the primary language of science for centuries, its relatively simple inflectional structure and limited vocabulary allow researchers to maintain simplicity in their models by relying on word-based assumptions and English datasets. However, it is important to note that even word-based models have not been considered scalable since the introduction of Mikolov's SkipGram model in 2013 (Mikolov et al., 2013a). Given these challenges, it is understandable that the DS community has largely continued to develop models at the word level.

Unfortunately, word-based models perform poorly on morphologically rich languages such as Turkish, Finnish, and Czech in terms of both scalability and effectiveness. In highly inflectional languages like Turkish, a single lexeme (root word) can generate thousands of surface forms (Sproat, 1992), leading to a strong possibility that a trained model will encounter an unseen, highly inflected word during testing, known as the *out-of-vocabulary (OOV)* problem. Turkish, for example, has a high morpheme-to-word type ratio (more than 3) (Oflazer, 1996; Jurafsky, 2000; Sak et al., 2012), where most morpheme transitions only slightly change the root meaning, rather than being derivational. Given this, it is reasonable to expect better results from distributional semantic models (DSMs) designed at the morpheme level for the Turkish language. Even if one attempts to mitigate the OOV problem by using an extremely large dataset (assuming scalability is not an issue), the quality of the learned vectors would likely be poor due to weak associations between infrequent, highly inflected word forms and their contexts, a challenge commonly known as the *rare-word* problem. Most word-based DSM studies (Mikolov et al., 2013a; Levy et al., 2015; Baroni et al., 2014) have reported improved model quality by filtering out words below a certain frequency threshold, assuming these words are either exceptional, proper nouns, or uninformative. Our own corpus coverage analysis reveals that 47% of

word types (277K) appear only once in the corpus, consistent with Sak's findings on the Boun Corpus coverage statistics (Sak et al., 2011). However, filtering out rare-words is not a viable option for Turkish, as it would eliminate more than half of the word types in the entire corpus, significantly reducing the model's lexical and semantic diversity.

Using sub-word based models (e.g., character, morph, or morpheme) is not a new idea in NLP research, particularly for morphologically rich languages. A notable example for Turkish is Sak's morpheme-based 'morpholexical' language model (LM) (Sak et al., 2012), which outperformed word n-gram based and statistical sub-word based LMs by leveraging Turkish morphology as a prior knowledge source. In the field of distributional semantics (DS), since 2013, several promising sub-word level DS models have been proposed to address the OOV and rare-word problems, sometimes referred to as morphological word embeddings. Most of these models are language-independent, statistical sub-word approaches that do not rely on any language-specific features (Grave et al., 2018; Luong et al., 2013; Botha and Blunsom, 2014; Cui et al., 2015). While some language-specific, morpheme-based (grammatical) models exist (Qiu et al., 2014; Bian et al., 2014; Lazaridou et al., 2013), they primarily focus on widely spoken languages such as English, German, and French. There are also studies focused on higher-level NLP tasks for Turkish (Kalender and Korkmaz, 2017; Yıldız et al., 2016; Demir and Özgür, 2014; Okur et al., 2016), which use word embeddings as an additional knowledge source.

This thesis aims to make conceptual assumptions and developments on different types of semantic relationships (such as relatedness and similarity), to model the role of incorporating language morphology as input in subword Distributional Semantics (DS) models, and to develop methodologies for generating the necessary datasets and evaluation methods to measure this impact. Within the scope of the study, static (non-contextual) embedding models and segmentation methods were empirically tested, the AnlamVer and OSimUnr word-pair datasets were produced, and a *relatedness-classification* task, along with related measurement methods was proposed to measure **the noise**

introduced by segmentation into the model. Our experiments demonstrate that morphological segmentation produces significantly less noise compared to n-gram-based methods and can provide substantial performance improvements depending on the nature of the task.

1.2 CONTRIBUTIONS

The main contributions of this thesis are as follows:

- Construction of AnlamVer[1], a word similarity and word relatedness evaluation dataset for Turkish.
- A new approach to analyzing and visualizing word similarity and relatedness data, featuring bi-dimensional values for each word pair.
- Key design insights for building a dataset that ensures a balanced representation of words and word pairs across various morphological and semantic characteristics.
- Construction of a publicly available[2] word relatedness dataset OSimUnr consisting of 372,559 word-pairs for Turkish and 639,993 for English, which focuses on special case *orthographically-similar-but-semantically-unrelated* word-pairs.
- Development of an open-source[3] dataset construction tool, including orthographic similarity and WordNet algorithms, and an English morphology stack. This resource can facilitate similar dataset generation and morphology modeling research for additional languages supported by NLTK, MorphoLex, Pyphen, and other resources.
- Development of grammatical DSMs for Turkish, exploiting Turkish morphology as a prior knowledge source to address OOV and rare-word problems, and reporting state-of-the-art results on the AnlamVer word relatedness dataset for Turkish with morphological segmentation. This

[1] http://www.gokhanercan.com/anlamver
[2] https://www.github.com/gokhanercan/OSimUnr or http://gokhanercan.com/OSimUnr
[3] https://github.com/gokhanercan/OSimUnr-Generator or http://gokhanercan.com/OSimUnr-Generator

includes benchmarking against existing literature as well as evaluating different word segmentation strategies, such as hyphenation, root-only and full morphological segmentation.

- Empirical evidence showing that FastText character n-gram based segmentation generates noise in semantic spaces, poses sensitivity to orthographic similarities of words which makes models unable to distinguish orthographically-similar words.
- Reporting the performance of morphological segmentation using the FastText model, which effectively addresses the noise issues caused by orthographic similarities. This approach achieves significantly higher accuracy (en = 68%, tr = 71%) while maintaining strong performance on conventional word relatedness benchmarks such as RareWords, MTurk771, MEN, and AnlamVer.
- Proposal of unrelatedness identification and relatedness-classification tasks, which provide insights into measuring the distinguishing ability of models, and experimentation with the task using various word segmentation settings.
- Benchmarks of WordNet-based relatedness/similarity approximation algorithms on word similarity datasets and the proposed tasks.
- Development of a methodology for applying fully derivational morphology (reducing to atomic roots) for both English and Turkish by integrating human-annotated resources (WordNet and Morpholex) with real-time morphological analysis and disambiguation tools.
- Compilation of an unlabeled Turkish corpus (tokenized and sentence-split) containing 5 billion words. The final corpora used in experiments for both English and Turkish exhibited vocabulary sizes exceeding five million unique tokens (en = 5.5M, tr = 5.2M).
- A publicly available web-based word similarity questionnaire software WSQuest.[4]

[4] http://gokhanercan.com/wsquest

Part of this thesis (ideas, tables, figures, results and discussions) have appeared or scheduled to appear previously in the following publications:

- Ercan, G., & Yıldız, O. T. (2025). Grammar or crammer? The role of morphology in distinguishing orthographically similar but semantically unrelated words. IEEE Access, 13 https://doi.org/10.1109/ACCESS.2025.3352186
- Ercan, G., & Yıldız, O. T. (2018). AnlamVer: Semantic model evaluation dataset for Turkish – word similarity and relatedness. In *Proceedings of the 27th International Conference on Computational Linguistics* (3819–3836).
- Yıldız, O. T., Avar, B., & Ercan, G. (2019). An open, extendible, and fast Turkish morphological analyzer. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)* (1364–1372). Varna, Bulgaria.
- Arıcan, B. N., Kuzgun, A., Marşan, B., Aslan, D. B., Sanıyar, E., Cesur, N., Kara, N., Kuyrukçu, O., Özçelik, M., Yenice, A. B., Doğan, M., Oksal, C., Ercan, G., & Yıldız, O. T. (2022). Morpholex Turkish: A morphological lexicon for Turkish. In *Proceedings of the Globalex Workshop on Linked Lexicography within the 13th Language Resources and Evaluation Conference (LREC 2022)* (68–74).

1.3 STRUCTURE

Structure of this thesis is as follows: Chapter 2 provides background information on semantics and distributed word representation modeling, evaluation methods, and common challenges. Chapter 3 discusses conceptual assumptions about semantic spaces concerning relatedness, semantic similarity, and orthographic similarity, with a focus on the Sim-Rel and OSim-Rel vector spaces developed in this thesis. Chapter 4 covers the distinctive properties of Turkish language, including morphological analysis and disambiguation methods. It also presents a fully derivational (with prefixes, suffixes and multiple

roots) morphological model, featuring the development of a lexicon with roots and words, a derivational database called MorphoLex Turkish, and a special word-level disambiguator. Chapter 5 describes the pipeline design considerations and methodology behind the construction of the AnlamVer and OSimUnr datasets developed in this study. Chapter 6 covers the experiments, including model configurations, segmentation methods, hyperparameters, metrics, corpora, visualizations, and results, along with comparisons to existing literature. Finally, Chapter 7 presents interpretations of the study's findings, along with discussions and conclusions that are addressed throughout this thesis.

CHAPTER 2

2. BACKGROUND

2.1 DISTRIBUTED WORD REPRESENTATIONS

2.1.1 Distributional Semantics

At the heart of distributional semantics (DS) research lies the idea that statistical distributions of linguistic items can help infer meanings. Turney and Pantel (2010) in their comprehensive paper "From Frequency to Meaning", describe this intuition as the *statistical semantics hypothesis*, stating: "statistical patterns of human word usage can be used to figure out what people mean." Indeed, machines lack true *understanding* of language. Language and meaning are subjective social constructs (De Saussure et al., 2011). Nevertheless, the statistical semantics hypothesis enables researchers to make more precise and practical assumptions about the specific tasks they are tackling. As *toolsmiths* (Brooks Jr, 1996), computer scientists have developed highly useful tools (e.g., search engines, speech recognition, machine translation) to meet human needs by making black-box statistical modeling assumptions rather than resolving the entire underlying structure of complex phenomena (Norvig, 2011).

2.1.2 Types of Distributional Relations

Linguists have been studying on distributions of linguistic items for a century. Although, distributional hypothesis—"words that occur in similar contexts, tend to have similar meanings"—is commonly attributed to Harris (1954), Sahlgren (2006) notes that the theoretical foundations of this methodology trace back to the structuralist linguists Bloomfield (1887 - 1949) and Ferdinand de Saussure (1857 - 1913). Saussure emphasized that *signs* within the language system can have distinctive functional roles. He identified two

(orthogonal) types of functional differences in linguistic elements, which are widely studied in DS research today: *syntagmatic* and *paradigmatic* relations (Sahlgren, 2006). In brief, "words have a syntagmatic relation if they co-occur, and a paradigmatic relation if they share same neighbors" (Sahlgren, 2006).

2.1.2.1 Syntagmatic relations

A syntagmatic relation defines the relationship as the occurrence of different linguistic items (entities) in similar contexts at the same time. Linguistic items with a syntagmatic relation occur sequentially, one after the other, and can belong to different parts of speech (POS) categories. Consider the sentence in the first row of Table 2.1, "Amerika savaş açtı" (America levied war), which consists of a proper noun, a noun, and a verb. A syntagmatic relation exists between the neighboring words *savaş (war)* and *açtı (levied)*. The co-occurrence of these two words is likely to happen more than once in similar contexts (e.g., world politics, military) like this. Such syntagmatic relations in word similarity evaluation tasks are classified as **relatedness** (i.e., semantic relatedness) (Finkelstein et al., 2001).

2.1.2.2 Paradigmatic relations

The word *paradigm* means an example, a model, or a prototype. A paradigmatic relationship refers to *substitutable* (prototypical) entities within a context. Unlike syntagmatic relations, words with paradigmatic relations do not occur simultaneously within the same context. However, they tend to appear in similar contexts and share common neighbors, with one instance often substitutable for another. Substitutable words are most likely in the same POS tag.

Paradigmatic relations of words can be observed in Table 2.1 at the column level. Since the words in the same columns are substitutional, alternative sentences (likely grammatically valid) can be constructed by replacing a word with its alternatives from the same column. In column 2, the relationship between *Amerika (America)* and *A.B.D. (U.S.A.)* is a good example of

synonymy. Conversely, the paradigmatic (horizontal) relationship in column 3 between *savaş (war)* and *barış (peace)* exemplifies an *antonymy* relation, which represents the relationship between words with opposite meanings.

In word similarity evaluation tasks (Finkelstein et al., 2001), both synonymy and antonymy relations are classified as similarity (i.e., semantic similarity) relations. Although most DSM studies tend to train and evaluate on similarity and relatedness, some studies (Agirre et al., 2009; Kiela and Clark, 2014; Hill et al., 2016) have attempted to distinguish these two concepts from each other.

Unfortunately, the distributional hypothesis is not sufficient to distinguish all semantic relations between words. Unsupervised semantic analysis models struggle to precisely detect semantic relations such as antonymy, hyponymy, and meronymy. These models typically capture only general relatedness and similarity scores. Table 2.2 presents some potential mappings of distributional relation types to semantic relations, as defined in the widely used lexical database WordNet (Miller et al., 1990). For example, in the case of hyponymy (e.g., *tree – plant*), it is unclear whether this relation should map to a syntagmatic relation, a paradigmatic relation, or both.

Table 2.1 Orthogonality of syntagmatic and paradigmatic relations. Adapted from Sahlgren's work (Sahlgren, 2006).

	Paradigmatic relations			Gloss
Syntagmatic relations	Amerika	savaş	açtı	America levied war.
	A.B.D.	barış	sağladı	U.S.A. has provided peace.
	Ülke	ateşkes	açıkladı	Country declared ceasefire.

A syntagmatic relation defines the relationship as the occurrence of different linguistic items (entities) in similar contexts at the same time. Linguistic items with a syntagmatic relation occur sequentially, one after the other, and can belong to different parts of speech (POS) categories.

Table 2.2 Mapping WordNet semantic relations to distributional relations and semantic similarity evaluation question types.

Semantic Rel.	Distributional Relation	Example	Semantic Eval.
synonymy	paradigmatic	sad, unhappy	similarity
antonymy	paradigmatic	war, peace	similarity
hyponymy	paradigmatic, syntagmatic	tree, plant	relatedness
troponomy	paradigmatic, syntagmatic	march, walk	relatedness
meronymy	syntagmatic	building, doors	relatedness
entailment	syntagmatic	sleep, snore	relatedness
syntactic			

2.1.3 Linguistic Regularities

The analysis of distributional relations of linguistic elements can capture two types of regularities in a language: **semantic** and **syntactic**. Semantic regularities refer to meaningful relations between entities and concepts in the real world. For example, *Ankara → Turkey* is an instance of the being-capital-city-of relation, representing the relationship between real-world entities. Similarly, *king → queen* or *cat → dog* are examples of the female-of relation, which represents more abstract concepts related to the real world.

In contrast, syntactic regularities describe the relationships between entities within the language itself, rather than the real world. These regularities often reflect the morphological (grammatical) rules of a language. For instance, *run → running* and *go → going* exemplify the gerund-form-of-a-verb (-ing) relation in English, which can be easily explained and evaluated through morphological knowledge. Syntactic regularities captured from DSMs can be used to induce morphological rules in a language in an unsupervised manner (Soricut and Och, 2015). One of the de facto standard DSM evaluation task, "word analogy" (Mikolov et al., 2013a), supports syntactic questions like "What is X to best as short is to shortest?" Figure 2.1 illustrates how the comparative-form-of and superlative-form-of syntactic regularities are captured by the GloVe DSM (Pennington et al., 2014). It’s important to note that the line between

syntactic and semantic relations is not always clear, as irregular or non-grammatical instances can occur, such as in the example *good* → *better* → *best*.

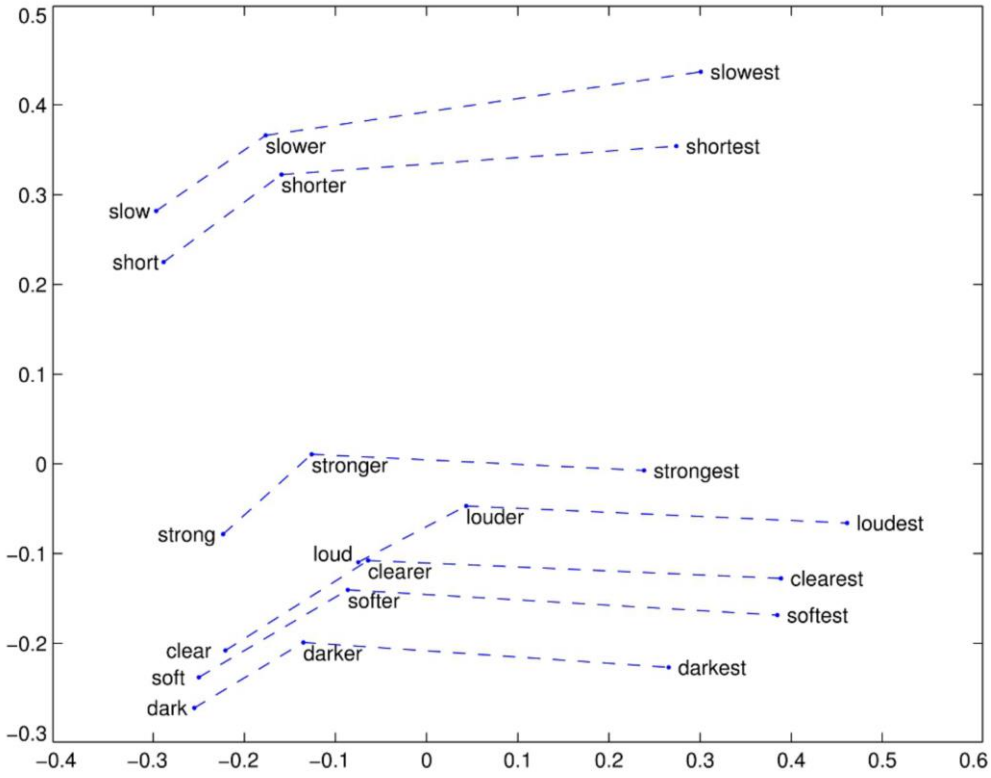

Figure 2.1 Figure taken from GloVe model's (Pennington et al., 2014) online documentation.

2.1.4 Vector Semantics with Static Embeddings

Representing co-occurrence information of linguistic items through vectors is commonly referred to as vector semantics. The main intuition is derived from the widely known vector space model (VSM) (Salton et al., 1975), which has been used in information retrieval (IR) tasks since 1975. VSM relies on the *bag-of-words* assumption. In this model, given a matrix where each column represents a document and each row represents a word, the **bag-of-words** approach assumes that by counting the frequency of each word's occurrence in documents, spatially closest document vectors will correspond to similar documents. The bag-of-words assumption simplifies IR models by

representing documents as a bag (multiset) of words, entirely ignoring the grammar and sequence of words in the text and focusing solely on word frequencies. VSM is one of the simplest yet foundational concepts behind IR, especially for search and indexing tasks. This approach forms the basis of **static embeddings**, where each word is represented by a fixed vector regardless of its context, in contrast to *contextual embeddings* (Qiu et al., 2020). Given their substantial complexity, contextual embeddings are excluded from the scope of this thesis, as our focus is on investigating the role of morphology and semantic relationships. In their paper, Arora et al. (2020) demonstrate that simpler, non-contextual embeddings (e.g., Word2Vec, GloVe, FastText) can perform within 5 to 10% accuracy of contextual embeddings on benchmark tasks, without incurring the orders of magnitude more computational costs typically associated with models like BERT (Devlin, 2018).

2.1.4.1 Choosing the Context

The basic representation of a VSM is a |W| × |C| co-occurrence matrix, where |W| is the number of words and |C| is the number of documents. This configuration is ideal for a document retrieval system, often referred to as a term-document matrix. Over 40 years of vector semantics evolution, the |W| dimension has remained constant, but the |C| dimension has varied depending on the task and model assumptions. The |C| context columns can be viewed as features in a machine learning model for each vector w in |W| (each row). From a feature selection perspective, it is crucial to select the most informative features and eliminate uninformative ones to mitigate the dimensionality problem.

In a semantic model, a typical configuration is a raw |W| × |W| matrix, where frequencies are collected by counting the occurrences of a word w within a window of n words to the left or right of the word, respectively. This approach is commonly known as the context window method and remains one of the primary hyperparameters in static word-embeddings. Depending on the model's assumptions, the window size n and its direction (left, right, or center) can vary,

and the window itself can represent a phrase, paragraph, sentence, or even a document.

Table 2.3 shows a sample context window where the window size n equals 7. The focus word *word* at index w_i is surrounded by $(n - 1)/2$ words on the left and $(n - 1)/2$ words on the right. Some studies suggest that larger window sizes may better capture relatedness, while models with narrower windows are more effective at reflecting similarity (Agirre et al., 2009; Kiela and Clark, 2014).

Table 2.3 Sample context window of size 7.

	w_{i-3}	w_{i-2}	w_{i-1}	focus word (w_i)	w_{i+1}	w_{i+2}	w_{i+3}		
You	shall	know	a	word	by	the	company	it	keeps.

2.1.4.2 Sparse and Dense Matrix Representations

Following the same |W| × |W| matrix M collected using the context window method, Table 2.4 illustrates what the values in a word-word co-occurrence matrix might look like. Given that |W| is large, it's unsurprising that the matrix would contain many uninformative zeros. M serves as a good example of a sparse matrix representation, which is sometimes referred to as a *one-hot vector* representation.

Table 2.4 Sample sparse matrix (M) representation.

M	w1	w2	w3	.	.	.	.	.	w|W|
w1	0	1	1	0	0	0	0	0	0
w2	1	0	0	0	0	0	0	0	0
w3	1	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
w|W|	0	0	0	0	0	0	0	0	0

Sparse matrix representations have two significant drawbacks:

- Dimensionality: Since both dimensions are linearly dependent on |W|, the memory and computational complexity of analysis techniques become very high. Sparse matrix representations are not scalable.
- Lack of Generalization: One-hot vectors lack generalized information (abstract concepts) because they do not share any common features other than co-occurrence information with the words. As a result, they are more prone to overfitting compared to dense vectors (Jurafsky, 2000; Levy et al., 2015). More generalized models tend to perform better in scenarios involving rare-words. For example, a sparse matrix may fail to capture synonymy relations because the vectors for *car* and *automobile* are represented by distinct dimensions (Jurafsky, 2000). In a rareword scenario where automobile occurs three times and car occurs 100 times, an overfitted model would be less likely to infer that the two words are similar through their shared contexts. Sparse matrices cannot leverage the continuity property of vector representations, causing two-word vectors to act as discrete representations since they do not intersect through a common context in the vector space.

2.1.4.3 Reducing Dimensionality

The columns of M can be considered features of word vectors. If a semantic model can be represented by the most informative features with a minimal size k, it can achieve better generalization and scalability. The process of selecting the best |C| features with size k from matrix M can be viewed as applying a mapping function g(M), which transforms the original sparse matrix M = |W| × |C| into a more compact matrix M′ = |W| × k, where k < |C| (Turian et al., 2010). DSM researchers have been searching for the optimal g(M) function that produces a denser, more informative matrix M′ with reduced memory and computational complexity, while also enhancing generalization.

2.1.5 Count-Based Distributional Semantic Models

A crucial aspect of DSM research is determining the best scalar values (measures) to populate the |W| × |C| matrix, which represents how likely words are to co-occur in specific contexts. Experience has shown that raw frequency values do not adequately capture these associations (Jurafsky, 2000; Baroni et al., 2014). To better represent word-context associations, raw frequencies often require information-theoretic smoothing and normalization transformations, similar to the widely used TF/IDF (Sparck Jones, 1972) formulation applied to term-document matrix frequencies in the field of IR. Models that utilize these raw or transformed frequency-based matrix representations are classified as count-based (i.e., traditional) models, as discussed in two extensive DSM evaluation papers by Baroni et al. (2014) and Levy et al. (2015).

2.1.5.1 Pointwise Mutual Information (PMI)

Pointwise Mutual Information (PMI) (Fano and Wintringham, 1961) is a widely used weighting scheme designed to represent the mutual information between words and contexts based on probabilities, assuming maximum likelihood estimation. PMI measures how often a word w and a context c co-

occur (joint probability) compared to the scenario where c and w are completely independent (marginal probability), as follows:

$$PMI(w,c) = log_2 \frac{P(w,c)}{P(w)P(c)}$$

A common technique is to replace negative PMI values, which represent very low co-occurring pairs, with zero—this is known as Positive PMI (PPMI):

$$PPMI(w,c) = max(log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$$

PPMI assumes there is little value in distinguishing the probabilities of very infrequent word-context pairs from each other (e.g., 10^{-12} vs. 10^{-11}) unless the corpus is enormous. It has been shown that PPMI outperforms PMI in semantic tasks (Bullinaria and Levy, 2012). However, one known drawback of PPMI is its bias toward infrequent events (Turney and Pantel, 2010), which in the DSM domain corresponds to rare-words. For example, if a very rare context c co-occurs with a word w only once, due to P (c) in the denominator of the formula, it would yield a relatively high PPMI score for that pair. To address this issue, Levy et al. (Levy et al., 2015) proposed the context distribution smoothing (CDS)[5] method, which mitigates this problem by replacing P (c) in the denominator with a modified function Pα(c), where the context probabilities are raised to the power of α as follows:

[5] Originally, same constant (hyperparameter) is proposed in Mikolov et al.'s SkipGram model. Levy et al. generalized the idea by showing it is applicable to other models too.

$$PPMI_\alpha(w,c) = max(log_2 \frac{P(w,c)}{P(w)P_{\alpha(c)}}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

Levy et al. showed that the configuration α = 0.75 does not only work for SkipGram, but also improves the performance of PPMI- and SVD-based models.

2.1.5.2 Singular Value Decomposition (SVD) Models

One common method for reducing the dimensionality of count-based models is applying SVD to their sparse matrices. Reducing the |C| dimension improves scalability and generalization power; *truncated SVD* is used to factorize the matrix into its most informative dense form for a given k. Since 1988, truncated SVD has been successfully applied to term-document sparse matrices as part of latent semantic indexing or latent semantic analysis (Deerwester et al., 1990). By selecting the top k dimensions where data varies the most, the resulting dense vectors capture *latent* features of the real world. Similarly, SVD factorization has become a key transformation method, M′ = g(M, k) = svd(M, k), applied to the M = |W | × |C| term-context matrix populated by the $PPMI_\alpha$(w, c) measure, where k serves as a model hyperparameter defining the new context size.

Although PPMI-based SVD methods (i.e., reduced PPMI) perform well, they have never been scalable due to the need to fully allocate the initial sparse matrix in memory. To mitigate this issue, several incremental approaches for constructing M ′ have been proposed, such as random indexing (Sahlgren, 2005) and random projection (Rehurek and Sojka, 2010). However, these methods have not gained much attention, as the focus has shifted to state-of-the-art prediction-based methods, such as Mikolov et al.'s SkipGram and CBOW (Mikolov et al., 2013a) models, and Pennington et al.'s GloVe model (Pennington et al., 2014).

2.1.6 Prediction-based Distributional Semantic Models

Prediction-based models gained significant attention following Mikolov et al.'s neural network-based SkipGram model in 2013. The key advantage of these models (i.e., context-predicting models) is straightforward: they avoid constructing a sparse matrix entirely. Instead, prediction-based models generate dense matrix representations directly, rather than reducing sparse matrices to dense ones. These models are trained similarly to traditional supervised learning tasks–now referred to as **self-supervised** (LeCun and Misra, 2021)–utilizing large quantities of both positive and negative samples, without incurring additional costs for human supervision. The goal is to maximize the probability of each context c, adhering to the same distributional assumptions about word-context co-occurrences as count-based models. Empirical results show that prediction-based models, particularly SkipGram and CBOW, outperform count-based models by a wide margin in terms of scalability (Baroni et al., 2014; Levy et al., 2015).

2.1.6.1 SkipGram Model

The SkipGram model is a prediction-based DSM featuring a shallow neural network architecture, inspired by intuitions from neural language modeling. It is best known for its open-source implementation library, *word2vec*[6]. SkipGram functions as a log-linear classifier that maximizes the prediction of surrounding words within a given context window. Probabilistic word and sentence prediction based on the local neighbors of a word has been successfully applied to LM tasks under the Markov assumption. SkipGram applies this same idea by treating the words within the window as positive and negative instances, learning weights (for k contexts) to maximize word predictions. During training, each word vector starts as a random vector and iteratively shifts towards its neighboring vectors. Each word vector has k dimensions (projection layer size), a model hyperparameter similar to the k in

[6] https://code.google.com/archive/p/word2vec/

the truncated SVD model. Figure 2.2 illustrates the single projection layer neural network architecture of SkipGram, along with the input and output matrices M and M′, which have dimensions of V × k and k × V, respectively. The objective of the SkipGram model is to maximize the average log probability:

$$\frac{1}{V}\sum_{v=1}^{V}\sum_{\substack{-n\leq j\leq n\\ j\neq 0}} \log\, p(w_{v+j}|w_v)$$

where V is the vocabulary size, w_1, w_2, ..., w_n is the sequence of training words, and n is the window size.

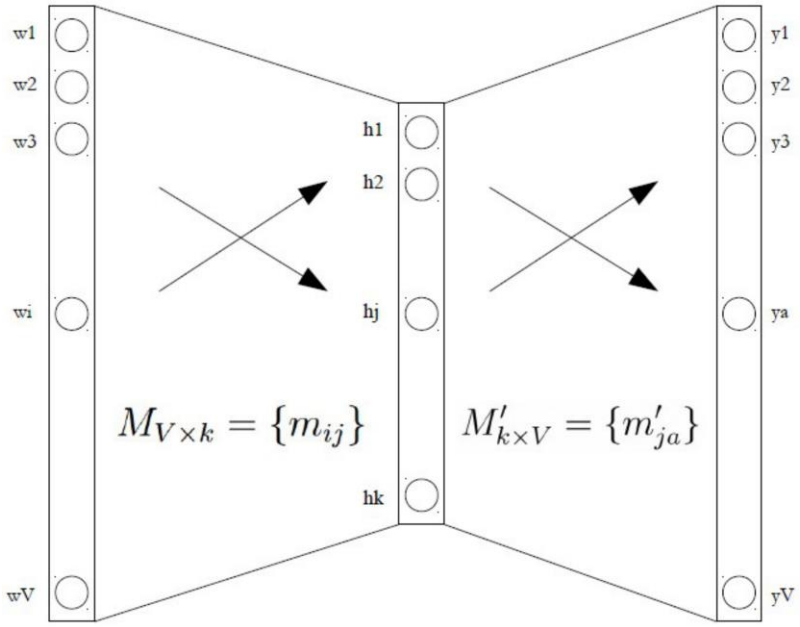

Figure 2.2 Neural network architecture of SkipGram model consisting of single projection layer.

Besides its scalability advantages and distinct neural network-based architecture, Levy et al. have shown that SkipGram's (SG)[7] objective function implicitly factorizes into a shifted PMI matrix, a well-known concept from the

[7] SGNS (SkipGram with Negative Sampling) was used in the original paper (Levy and Goldberg, 2014) instead of SG. The details of negative sampling are omitted here for clarity.

word-similarity literature (Levy and Goldberg, 2014). Thus, the information-theoretic foundation of SkipGram is arguably similar to that of count-based models:

$$M_{ij}^{SG} = W_i \times C_j = PMI(w_i, c_j) - log\, k$$

Moreover, Levy et al. wrote an extensive paper (Levy et al., 2015) demonstrating that certain hidden hyperparameter tunings and methods (CDS, sub-sampling, deleting rare words, negative sampling) in the SkipGram model can be applied to other models—both count-based and prediction-based—to improve their efficiency. Both Baroni et al.'s and Levy et al.'s extensive benchmarks agree that the SkipGram and CBOW models outperform other models (including GloVe) by a large margin in terms of scalability. However, while Baroni et al.'s report claims that SkipGram outperforms all other models in efficiency, Levy et al.'s benchmark suggests that no single model consistently outperforms others across all tasks and hyperparameter settings with a significant margin (Levy et al., 2015).

2.1.6.2 CBOW Model

The CBOW model, also proposed by Mikolov et al., is reported to be more scalable than the SkipGram model, though it incurs a slight efficiency loss on semantic tasks. As shown in Figure 2.3, while SkipGram predicts the surrounding words of a given word w, the CBOW model predicts w based on its context. Since the CBOW model ignores word order, Mikolov et al. (2013a) noted that it performs worse on semantic tasks. However, some studies report opposing results (Baroni et al., 2014; Levy et al., 2015).

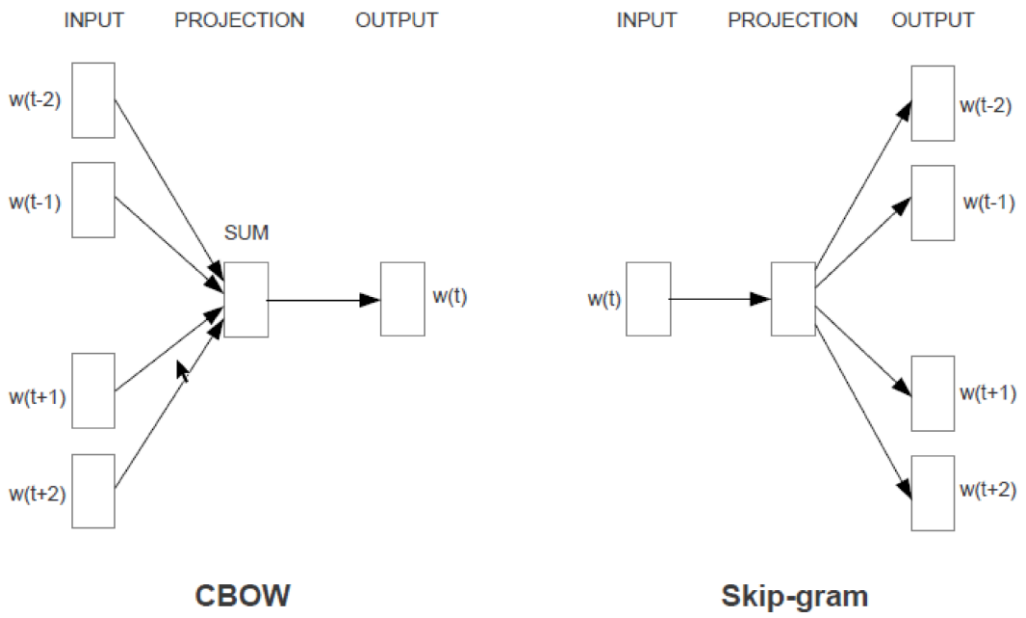

Figure 2.3 SkipGram and CBOW architectures. Taken from the paper by Mikolov et al. (2013a).

The SkipGram and CBOW models are scalable enough to train on 100 billion-word datasets in a single day, even on a single machine implementation. The complexity analysis in the original paper (Mikolov et al., 2013a) demonstrated that both models are linearly dependent on k (the projection layer size) and logarithmically dependent on the vocabulary size V. Unlike the CBOW model, the SkipGram model is also linearly dependent on the window size n.

$$Q_{sg} = O(n \times (k + k \times \log(V))) \tag{2.1}$$
$$Q_{cbow} = O(n \times k) + (k \times \log(V))$$

Formulation 2.1 demonstrates the complexity of the two objectives, where n is the context window size, k is the context dimensionality (projection layer size), and Q_{sg} and Q_{cbow} denote the computational complexity of the SkipGram and CBOW models, respectively. Empirical results show that training CBOW and SkipGram took 1 day and 3 days, respectively, on a 783M dataset (a subset of the Google News dataset) with n = 300 (Mikolov et al., 2013a).

2.2 ENRICHING WORD EMBEDDINGS

A key property of unsupervised representation learning tasks is that their output can be easily applied to downstream tasks—often referred to as pretrained language models. Distributional Semantic Model (DSM) outputs, such as word embeddings, have been successfully applied across a wide range of NLP tasks, enhancing model efficiency.

Since 2013, an increasing body of literature has focused on *enriching word embeddings* by incorporating prior knowledge. This enrichment can be semi-supervised when labeled data, such as POS tags, is employed. Broadly, two main categories of word embedding enrichment strategies exist: knowledge-based and morphology-based.

Knowledge-based word embedding enrichment methods typically use pre-existing semantic databases (lexicons) as prior knowledge, such as WordNet, FrameNet, the Paraphrase Database, and parallel corpora (Yu and Dredze, 2014; Faruqui et al., 2014; Bian et al., 2014). Conversely, word embeddings can also be used to induce morphology (Yıldız et al., 2016; Soricut and Och, 2015; Üstün and Can, 2016) or improve lexicons (Yu and Dredze, 2014; Bian et al., 2014; Faruqui et al., 2014). Additionally, some studies leverage language morphology (mostly POS tags) (Cotterell and Schütze, 2015) or global context (Huang et al., 2012) at the word level to enrich word embeddings. In this thesis, we focus on enrichment through morphological segmentation while leveraging WordNet as a knowledge base for relatedness approximation during evaluation.

2.2.1 Morphological Embedding Models

Word-based embedding models ignores the internal structure of words which reduces model capability and quality. More complex but efficient morphological word embedding models are proposed to overcome this problem. Morphological word embedding enriching methods fall into two main categories: *grammatical* and *statistical*. Almost all morphological enriching models are at sub-word level, either char, morph, or morpheme based. Main idea

behind morphological word embeddings is leveraging prior language information by analyzing internal structure of words to alleviate *out-of-vocabulary* (OOV) and *rare-word* problems.

While grammatical models are language-specific and tend to perform best within their target language, statistical models do not rely on language-specific features, making them applicable across multiple languages, which is a key advantage. We use the term **morpheme** for a meaningful grammatical unit that cannot be further divided, in contrast to morph, which refers to a statistically derived sub-word unit that may not be meaningful or grammatically correct. Table 2.5 summarizes the terminology and classification of these embedding models.

Table 2.5 Morphological and non-morphological segmentation types.

Word Level	Morphology	Sub-unit	Splitting
Word	no	word	-
Statistical (Sub-word)	no	morph	Word Segmentor
Grammatical (Sub-word)	yes	morpheme	Morphological Analyzer
Syllabification (Sub-word)	partially	syllable	Syllabification
N-gram (Sub-word)	no	n-gram	N-gramming
Char	no	char	Char Array

2.2.2 Statistical Embedding Models

Since statistical sub-word models do not rely on any language-specific features, they require unsupervised word segmentation algorithms to analyze inner word-level statistics. One popular algorithm for this task is Morfessor (Creutz and Lagus, 2007), which uses *minimum description length analysis* and has been shown to perform very well compared to widely known benchmark algorithms. However, an unsupervised segmentation of the word residing might produce the incorrect segmentation *+re+sid+ing*, whereas the correct analysis should be *_reside+ing* (root and suffix). As such, the unsupervised and language-independent nature of the Morfessor algorithm can result in incorrect

segments (morphs), which may reduce model performance compared to grammatical analyzers.

2.3 SUBWORD-LEVEL MODELING

Subword-level modeling has gained popularity in NLP research due to its ability to enhance generalization by leveraging subword-level information, thus aiming to free models from representing an infinite number of words. The underlying idea is simple: instead of learning the semantics of every single word in a language, models learn a finite number of subword-level units (e.g., morpheme, syllable, character n-gram, segment) and the rules governing their composition. This parallels the concept of deriving the meaning of any given sentence by composing the limited representations of its constituent words. Subword-level modeling takes this abstraction to the next level. Considering that languages have a limited number of lexical roots and affixes—MorphoLex English derivational database (Sánchez-Gutiérrez et al., 2018) represents ≈70K words with ≈15K roots (including some proper nouns) and 422 affixes—this concept initially sounds appealing. Assuming Zipf's law (Zipf, 1935) holds for languages, most units are very rare. Thus, correctly modeling a limited number of non-rare units might suffice to represent the entire language. However, as Anderson (1972) criticized constructionist hypothesis, "The ability to reduce everything to simple fundamental laws does not imply the ability to start from those laws and reconstruct the universe." Therefore we should be aware that composing a word from its constituents might not be as straightforward as segmenting it into them. On the other hand, although **systematic compositionality** of languages is questionable, "studies suggest that deep networks are capable of making subtle grammar-dependent generalizations" (Baroni, 2019). If such constructionism—where subword units systematically create meaningful words—is possible for languages, smaller models trained with smaller corpora could potentially overcome foundational challenges in NLP applications, especially the out-of-vocabulary (OOV) and rare-word problems.

These issues are particularly pertinent in morphologically rich languages such as Turkish, Czech, or Finnish.

2.3.1 Word Segmentation

Subword-level DSM consists of two important components: (i) a **word segmentation** method for splitting words into their constituent subword units, and (ii) a **modeling objective** for learning subword representations and composition rules among them. Word segmentation is one of the early and essential stages of the NLP pipeline because of its inherent reusability potential across many downstream tasks. It can vary in complexity, ranging from simple methods such as n-grams or hyphenation (i.e., syllabification) with very low costs, to more complex, such as morphology-aware or task-specific approaches. For optimal task performance, we believe that one or both components must exhibit sufficient complexity or customization tailored to the specific task at hand. One example of a simple segmentation is the Turkish syllabification, which has only four simple rules (e.g., (i) all syllables contain one vowel) (Göksel and Kerslake, 2004) to follow, which only takes 30 lines of implementation code without any training involved. Alternatively, it is possible to reuse resources generated by unsupervised statistical methods such as Morfessor (Virpioja et al., 2013), Byte Pair Encoding (BPE) (Gage, 1994), SentencePiece (Kudo and Richardson, 2018) or supervised segmentation methods such as CHIPMUNK (Cotterell et al., 2015). Finally, arguably the most costly option is **morphological segmentation**, which leverages prior morphological information to split words into morphological units called morphemes. Hence, choosing the best word segmentation method for a task remains an important question. As reported in the study by Zhu et al. (2019), no segmentation method (including morphological segmentation) consistently outperforms others, and ”performance is both language- and task-dependent.” It should be noted that approaching the text splitting problem at the *word* level is itself a presumption. For example, from a Zipfian point of view, according to the study by Ryland Williams et al. (2015), phrases obey Zipf’s law more closely

than words and other subword units, and they comprise the most coherent units of meaning.

2.3.2 Language-Independence

The choice of word segmentation method is influenced by several factors, with a significant consideration being whether to maintain models *language-independent* (i.e., language-agnostic) or not. FastText model (Bojanowski et al., 2017), which is used as a modeling tool in this study, sets a foundation for semantic modeling research by incorporating both simple word segmentation and fundamental modeling objectives CBOW and SkipGram. FastText extends the well-known Word2Vec (Mikolov et al., 2013c) model to subword-level by using character n-grams as a segmentation method, employing the same objectives. It produces subword-level static embeddings with notable training efficiency, making them highly reusable for various NLP tasks. There is no doubt that keeping models language-independent makes them easily reproducible across multiple languages. For example, two separate studies (Grave et al., 2018; Heinzerling and Strube, 2018) applied language-agnostic segmentation methods: n-grams and BPE, respectively, and released pre-trained embeddings for 157 and 275 languages using multilingual corpora such as Wikipedia and Common Crawl.[8] [9]

On the contrary, as Bender (2013) stated "knowledge of linguistic structure is crucial for feature design and error analysis in NLP", we generally assume that linguistic resources are beneficial. Language morphology, being a complex phenomenon, typically requires sophisticated models and substantial computational resources to learn from scratch. Despite the higher costs and language-specific constraints, incorporating prior linguistic knowledge into models is expected to enhance their performance in the target language compared to generic approaches. Thus, in theory, **handcrafting morphological information** could serve as a beneficial shortcut to improve model performance.

[8] https://github.com/bheinzerling/bpemb
[9] https://fasttext.cc/docs/en/crawl-vectors.html

More broadly, as Sutton (2019) argued, the human-knowledge approach (equivalent to incorporating language morphology in our context) is anticipated to make a difference at least in the short-term compared to the ultimate massive computation powered solutions.

2.3.2 The Role Morphology

Most subword-level DSM studies, however, have not shown any significant advantage of using linguistic knowledge as input over using language-independent methods, especially on **conventional wordsim** (i.e., word similarity/relatedness) tasks (Qiu et al., 2014; Bojanowski et al., 2017; Zhao et al., 2018; Romanov and Khusainova, 2019). A study by Zhao et al. (2018) shows that the statistical methods, such as BPE and Morfessor, cannot outperform the FastText n-gram benchmark on the Turkish word relatedness dataset AnlamVer (Ercan and Yıldız, 2018). They state that it is due to the *noise* generated from the syntactic affix concatenations in Turkish. Another study (Romanov and Khusainova, 2019) reports that ”the choice of subword units—morphemes or n-grams— doesn’t make much difference” on part-of-speech, Chunking, and NER tasks in Russian language. An earlier study (Qiu et al., 2014) shows that using morphological morphemes (especially roots) from Longman Dictionaries slightly improves performance on analogy and wordsim tasks. In their base model MorphemeCBOW and its variants, they customize the CBOW objective by adding auxiliary POS inputs and defining coefficients that differentiate roots from affixes in varying weights. However, even in the best cases, their improvements are limited to 2-3 percentage points compared to Morfessor and syllable segmentation (e.g., Root=43.29, Morfessor=40.32, Syllable=41.29).

In another work, Üstün et al. (2018) report a 5% improvement on the analogy task for Turkish with their morpheme segmentation model *morph2vec*, while they observe no improvements on the wordsim task using the English datasets RareWords and WordSim353 (FastText=0.529, morph2vec=0.38). In their paper, they state that ”*orthographic* commonness of words, that governs orthographically similar words to have similar word representations.” They also

emphasize that n-gram segmented spaces are affected by orthographic similarities (i.e., string similarity, spelling similarity, lexical similarity) of units. For Turkish, they report a significant performance improvement on the WordSimTr word similarity dataset they designed (FastText=0.208, morph2vec=0.529). This improvement can be attributed to the notably high average orthographic similarity of word-pairs (5.62/10) within the WordSimTr dataset, due to the inflectional nature of the selected word-pairs. Table 5.9 and Figure 3.4 show the average orthographic similarity values of word-pairs in some commonly used datasets along with our OSimUnr sub-datasets (Q3 and Q4). As a side note, possibly owing to the larger corpora we utilized, we achieved a higher benchmark score of 0.58 using FastText char-gram segmentation, whereas our morphological models attained 0.68 and 0.78 on the same WordSimTr dataset (Table 6.7).

We contend that the analogy and word similarity tasks, due to their *relative* querying natures, are not ideal for investigating the contributions of morphology to semantic spaces. By *relative* querying, we refer to queries such as "King is to X as Man is to Woman, find X" and "what is the ranking correlation between model predictions and human scores," which **do not involve real valued scores**. We also argue that the way we choose word-pairs in widely used wordsim datasets might be hiding contributions of such linguistic knowledge. Despite the declining popularity of the wordsim task in favor of more complex natural language understanding (NLU) tasks such as GLUE (Wang et al., 2018), MMLU (Hendrycks et al., 2020) or BIG-Bench Hard (Suzgun et al., 2022), we propose revisiting the word relatedness task from a new perspective, focusing on word-pairs that are **orthographically-similar-but-semantically-unrelated** (i.e., **OSimUnr**).

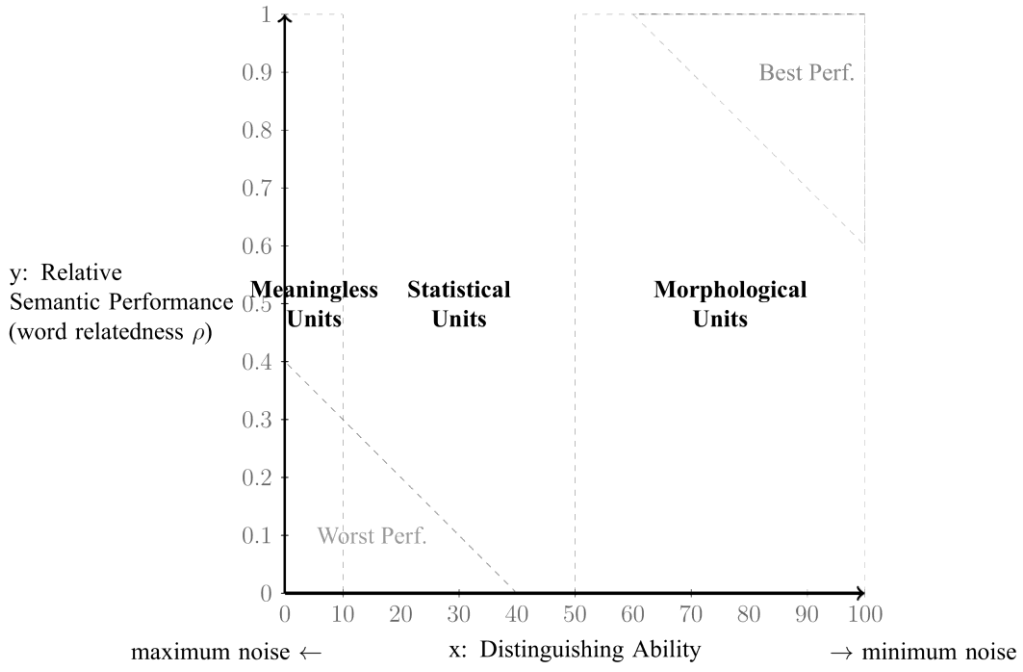

Figure 2.4 Semantic Clarity Space: A conceptual diagram illustrating how noise decreases as the meaninglessness of segmentation units decreases.

As the *grammar – crammer* word-pair exemplifies, a good semantic model should easily distinguish two unrelated words semantically, even if they are orthographically-similar. We accept that comparing two orthographically-similar words is an extreme case and it might not seem like a crucial problem for a regular downstream task at first glance. However, even if the word-pairs are not orthographically-similar, we show that evaluating the *ability of distinguishing concepts from each other* (i.e., distinguishing ability) of semantic models might be an insightful indicator due to its highly negative correlation with **the noise** generated by the segmentation methods (x-axis in Figure 2.4). That is why we propose the **relatedness-classification** and **unrelatedness-identification** tasks that measures the distinguishing ability of semantic models. We posit that measuring and improving that ability can be helpful in application-level tasks such as spelling correction, text simplification, or text generation. According to the definition provided by Bender and Koller (2020), *form* refers to "any observable realization of a language", while meaning pertains to

something external to language: "relation between the form and communicative intent". How can we advance when our models and evaluation methods lack the ability to distinguish between various forms?

For the empirical evaluation, we constructed a word relatedness dataset that contains only word-pairs that match the aforementioned OSimUnr special case conditions. We applied the same methodology to two structurally different languages, English (en) and Turkish (tr), to measure the impact of Turkish language's agglutinative, highly productive, and inflectional morphology against English language. Our experiments show that, regardless of the modeling objective, widely used character n-gram segmentation with the FastText model performs very poorly (below 5% accuracy) on the *unrelatedness-identification* task we propose (Table 6.4). Conversely, morphological segmentation overcomes the problem (en=68%, tr=71% accuracy) while performing similarly on conventional wordsim evaluations (Table 6.7, Figure 7.1). Table 2.6 shows some examples from the final dataset, demonstrating how morphological segmentation (FT-M and FT-MR) normalizes the poor estimations of the default n-gram segmentation (FT) when the word-pairs are orthographically-similar (OSim) but semantically unrelated (Rel).

Table 2.6 Cherry-picked examples from the final OSimUnr dataset.

Language	Word-pairs	OSim	Rel	FT	FT-M	FT-MR
English	grammar – crammer	0.71	0.22	0.45	0.19	0.25
	shrink – shrine	0.83	0.24	0.68	0.01	0.02
	internet – intercept	0.77	0.18	0.63	0.40	0.24
	adventure – denture	0.77	0.22	0.81	0.31	0.05
	fridge – fringe	0.83	0.24	0.72	0.23	0.26
Turkish	nakliyat – bakliyat	0.88	0.19	0.84	0.55	0.25
	çimenlik – çevirmenlik	0.73	0.22	0.61	0.55	0.01
	kampanya – şampanya	0.88	0.17	0.73	0.07	0.12
	indirme – sindirme	0.88	0.24	0.86	0.89	0.46
	bakara – makara	0.83	0.18	0.75	0.21	0.17

All values are normalized to [0-1] scale. OSim scores are calculated by editsim. Rel denotes WordNet relatedness approximations. FT: default FastText, FT-M: morphological segmentation, FT-MR: root-only morphological.

CHAPTER 3

3. SEMANTIC SPACES

3.1 RELATEDNESS AND SIMILARITY

The common assumption behind unsupervised DSM research is the **distributional hypothesis**, which states ”words that occur in similar contexts, tend to have similar meanings” (Harris, 1954). In the early years of NLP research, the phrase *similar meanings* led to terminological ambiguities among researchers. The terms *relatedness* (i.e., association) and *similarity* were used interchangeably. Consequently, most datasets' scores were collected by ambiguous annotation guidelines (Hill et al., 2016). Currently, a consensus has emerged to distinguish between the two terms as follows: while **relatedness** refers to any association between two concepts if they co-occur in the same context, regardless of their functional roles (e.g., *driving – car*), **similarity** (i.e., attributional similarity) refers to a *paradigmatic relation* (De Saussure et al., 2011) between concepts that share common properties and are likely to share same neighbors but while being substitutional in the same context (e.g., bike – bicycle). Even though there are no consistent exact definitions of such terms, recent datasets such as SimLex-999 (Hill et al., 2016), AnlamVer (Ercan and Yıldız, 2018), SuperSim (Hengchen and Tahmasebi, 2021), SimRelUz (Salaev et al., 2022) adhere to this distinction by displaying annotation guidelines to their participants with their own words and examples.

According to the results of the DSM studies that used those datasets (Zhu et al., 2019; Levy et al., 2015; Hengchen and Tahmasebi, 2021), we can generalize that modeling the similarity relation is more challenging (SimLex=0.28, AnlamVerSim=0.35) compared to modeling the relatedness relation with unsupervised DSM methods (WSRel=0.62, AnlamVerRel=0.45).[10]

[10] AnlamVer Turkish dataset includes two distinct scores for each word-pair which here

In their studies, Hengchen and Tahmasebi (2021) and Hill et al. (2016) conclude that modeling the relatedness is easier than modeling the similarity. Our word relatedness and similarity experiment (Table 6.7) also justifies this hypothesis on Turkish AnlamVer dataset (ρ_{sim} = 0.44, ρ_{rel} = 0.74), since the AnlamVer dataset contains both relatedness and similarity scores for every word-pair.

Furthermore, most word similarity datasets (e.g., SimLex-999, AnlamVerSim, SimRelUz) conventionally guided their annotators to score antonyms as ”dissimilar”, a practice that has been identified as a mistake. This should be the opposite from both distributional modeling and linguistic perspectives as discussed in the studies Kliegr and Zamazal (2018); Ercan and Yıldız (2018); Vulić et al. (2020). Similar to synonymy, *antonymy* is a paradigmatic type of relation that is highly substitutional. Antonym pairs are likely to share common attributes in a semantic network such as their POS. For instance, in the sentence ’Joe is very dumb | smart’ —which has score 0.75/10 in SimLex-999—two adjectives are substitutional and attribute to the same feature of Joe even though they change the meaning of the sentence in one dimension. This is clearly one of the reasons for low DSM scores on word similarity, even though DSMs are considered capable of modeling both syntagmatic and paradigmatic relations (Lapesa et al., 2014). Similarity datasets include such antonym pairs scored as *dissimilar* to a considerable extent (6% of SimLex-999, 10% of AnlamVerSim), which are inherently incompatible with DSMs and knowledge bases such as WordNet. In OSimUnr dataset, we focus on **relatedness** relation by using word relatedness datasets as primary indicators because this relation type is well studied, relatively easy to model, and inherently **compatible with the distributional hypothesis**. We treat all traditional wordsim datasets (e.g., MEN, WordSim353) as relatedness datasets because the annotators tend to score the relatedness of words instead of the similarity when there is no clear distinction is provided in the guidelines. We include some of the word *similarity* datasets SimLex-999, AnlamVerSim, and WordSimTr in our

referred to AnlamVerSim for similarity and AnlamVerRel for relatedness.

experiments for benchmarking purposes only. Therefore, we exclude them from aggregate results of the relatedness experiments (Table 6.7, 6.8).

3.2 THE NOISE

3.2.1 Shared Meaninglessness

Although they are easy to implement, simple segmentation methods such as character-grams or syllables generally segment words into meaningless units. They can only represent the original concept by concatenating atomic units by applying some combinational repetition mechanism with the cost of **the noise** it generates. For example, Char-gram[3-6] (i.e., CG[3-6])[11] segments the word *glowing* into 22 grams such as '*glo, glow, glowi, ..., win, low, lowi, ..., wing, ing*' as shown in Table 3.1. FastText is a bag-of-subwords model, where in the training phase, each of those sub-units are equally weighted within a context, such as 'He gave her a [*_glo, _glow, _glowi, ...,_low, ...,_wing, ...,_ing*] smile'. The problems with this training context are two-fold. First, every meaningful sub-unit such as *_glow* or *_ing*, can represent the concept glowing with a fraction of its full meaning, approximately 1/22 of its potential. Consequently, additional meaningless n-grams (e.g., *_glo,_owi*) are always necessary to construct the complete meaning of the concept. Secondly, the meaningless units lack linguistic (i.e., morphological) relevance, making them unlikely to occur systematically in related contexts, particularly in alphabetic languages. They are most likely to occur in unrelated contexts too, which adds lots of noise to the semantic space. As the noise increases, **everything gets more related to each other** to some extent. For instance, sub-units like *_win*, *_wing*, and *_low* also partially represent concepts like *win* (to win), *wing* (organ) or *low* (adjective) which should not be related with the glowing itself. In a noisy semantic space, even a random word-

[11] Char-gram[3-6] refers to all possible character-grams where minimum gram length is 3 (e.g., glo) and the maximum gram length is 6 (e.g., glowin). It is the default and the most used configuration of FastText. Square brackets indicates inclusion of word starting and ending symbols '<' and '>' (e.g., <glo).

pair like *lyqmsns* – *ashwnsuv* receives similarity score of 4/10, whereas morphological representations yield values close to zero (FT=0.40, FT-M=-0.05, FT-MR=-0.15).

Moreover, if frequencies of units matter in modeling a language, as reported by Ryland Williams et al. (2015), n-grams overlap in their counting, which ”obscures underlying word frequencies.” As the authors also state ”we are unable to properly assign rankable frequency of usage weights to n-grams combined across all values of n”, they don’t even come close to obeying Zipf’s law. Indeed, it is evident that **we sacrifice valuable word-boundary information** when transforming words into n-grams. To avoid losing that information, the FastText implementation also adds the surface form of the word itself (e.g., *glowing*) into the bag-of-units along with the n-grams, which can be only helpful for non-OOV cases (see first units of Char-gram[3-6] column in Table 3.1).

Table 3.1 An example from overlapping units of char-gram[3-6] segmentation.

Word	Char-gram[3,6] Segmentation	Morph. Seg.
glowing	glowing, <gl, <glo, <glow, <glowi, glo, glow, glowi, glowin, **low, lowi, lowin, lowing, owi, owin, owing, owing>, win, wing, wing>, ing, ing>, ng>**	glowing, _glow, **+ing**
slowing	slowing, <sl, <slo, <slow, <slowi, slo, slow, slowi, slowin, **low, lowi, lowin, lowing, owi, owin, owing, owing>, win, wing, wing>, ing, ing>, ng>**	slowing, _slow, **+ing**

The chars ’<’ and ’>’ indicate beginning and ending characters of words. Overlapping units are indicated in bold. Units that may possess alternative interpretations (e.g., wing, win, low) across various domains are underscored.

3.2.2 Overlapping N-grams Problem

Aside from the noise it generates, when it comes to the orthographically-similar word-pairs scenario, another problem *overlapping-n-grams* arises. As overlapping n-grams are highlighted in Table 3.1, if two words are orthographically-similar, most of their n-grams overlap with more than a half ratio (63.63% for *glowing* – *slowing*), meaning that those concepts are represented in the semantic space by the same vectors to that extent. Considering that bag-of-units models represent words by getting the average or sum of its

unit vectors (i.e., aggregate vectors), it is a big challenge for them to distinguish two concepts from each other. The overlapping factor of n-grams for shorter words is reasonably low (16.66% for *car – bar*), especially when the word lengths are lesser than the maximum n-gram value (default is 6 for FastText). However, due to the nature of the n-gram algorithm, as the lengths of the words increase, the overlapping factor increases linearly (Table 3.2). It is important to acknowledge that the highest degree of overlapping occurs when the character changes are located around the starting or ending regions of the words. To measure the character differences (edit distance) of word-pairs, we can employ the well-known edit distance algorithm introduced by Levenshtein (1966). When the edits are in the middle, and the word lengths are short, the overlap is relatively lower (22.22% for *fridge – fringe*). But when the one edit distance is on the first or last character, for highly derivational and/or inflectional cases like *tencerelerimizden – pencerelerimizden*, the overlapping factor can be as high as 87.09% even though their lexical roots are totally unrelated (*_tencere* [pot] – *_pencere* [window]). That level of suffixation is not an extreme case for an agglutinative language such as Turkish.

Table 3.2 Linear relationship between word lengths char-gram overlaps. All edits are in the first letter of words. See Equation 3.3 for editsim formulation. FT and FT-MR relatedness scores are normalized to [0-1] scale. Len=word length, CG=Char-gram[3,6].

One edit distant word-pair [lang]	Len	CG	OL#	OL%	editsim%	FT	FT-MR
car - bar [en]	3	6	1	16.66	66.66	0.32	0.27
tablo - kablo [tr]	5	14	6	42.85	80.00	0.50	0.15
glowing - slowing [en]	7	22	14	63.63	85.71	0.88	0.29
biracılık - kiracılık [tr]	9	30	22	73.33	88.88	0.90	0.25
mindlessness - windlessness [en]	12	42	32	80.95	91.66	0.96	0.24
tencerelerimizden - pencerelerimizden [tr]	17	62	54	87.09	94.11	0.98	0.42

3.2.2.1 Orthographic Similarity Correlation Problem

Unlike the orthographic similarity algorithms, semantic models should distinguish unrelated word-pairs by their meanings regardless of their

orthographic resemblance or overlapping factor of units. However, our analysis reveals a **strong positive correlation** (up to ρ = 0.50) between FastText's predictions and the orthographic similarity scores of orthographically-similar word-pairs. Figure 3.1 shows some of the orthographic similarity algorithms correlate with FastText's predictions (FT-CG) regardless of the language and the sub-dataset type Q3 or Q4. We contend that the default Char-gram segmentation is the underlying cause of this **orthographic sensitivity**. Our morphologically segmented model FT-M and FastText model FT-CG demonstrate relatively low correlation (0.291 and 0.164, respectively), despite being trained with the same objective and hyperparameters, differing only in the segmentation. The figure also shows that our FT-M model exhibits no correlation with orthographic similarity algorithms.

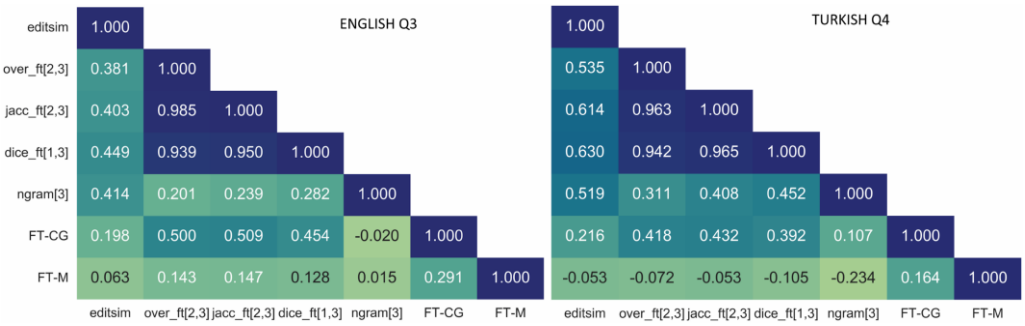

Figure 3.1 Spearman correlations (ρ) of similarity scores for semantic models and orthographic similarity algorithms. Experiments ran on OSimUnr editsim dataset. Q3 word-pairs are sampled to 20,000 items. Dice, Jaccard, Overlap coefficients are calculated using FastText n-grams. Ngram[3] (i.e., ngr3) denotes n-gram similarity algorithm. See §3.4.1 for the algorithms.

3.2.2.2 The Noise Across Linguistic Typologies

From a linguistic perspective, we can generalize that as the average number of morphemes per word (i.e., the index of synthesis (Karlsson, 1998)) progressively increases from isolated to fusional, agglutinative, and polysynthetic languages, the severity of the aforementioned n-gram issues becomes more pronounced. For instance, in isolating and analytic languages

such as Chinese and Vietnamese, most words are either monomorphemic or consist of two morphemes, resulting in an index of synthesis close to zero (see Table 3.3). For example, the index of synthesis for Turkish (tr = 2.86) is nearly double that of English (en = 1.68). The noise introduced by n-gramming might exhibit a positive correlation with the index of synthesis.

Table 3.3 Index of Synthesis

Language	Index of synthesis
Vietnamese	1.06
Yoruba	1.09
English	1.68
Old English	2.12
Swahili	2.55
Turkish	2.86
Russian	3.33
Inuit (Eskimo)	3.72

Table taken from Karlsson (1998) (Karlsson, 1998).

Similar to the synthetic levels of languages, writing systems also play a significant role. As the essential unit of writing systems transitions directionally from representing an idea (pictographic) to a morpheme or word (logographic), to a syllable (syllabic), and ultimately to a sound (alphabetic), the degree of abstractness and meaninglessness of the units progressively increases. In alphabetic systems such as English and Turkish, sounds—which are inherently meaningless—are represented by letters. This approach results in a limited set of alphabetic characters but leads to greater repetition and more meaningless combinations in written forms. In Chinese, a logographic language, representing a word with a logograph is efficient from an information-theoretic perspective but results in a writing system with a vast number of symbols. This can be seen as an advantage from a modeling perspective, as it eliminates the need for subword modeling and introduces less noise. However, it is ultimately a trade-

off. This approach reduces the channel capacity while increasing the vocabulary size, which can limit the creativity and reusability of blocks—for example, in forming new concepts—to some extent. Modern Chinese, for instance, uses thousands of characters, with approximately 3,500 required for basic literacy and around 8,000 for advanced literacy.

Another dimension on the effect of the language structure is the alphabetic languages ability to have clear morpheme boundaries. As a canonical example, Turkish, an *orthographically transparent* language, is written as it is pronounced, exhibiting a consistent and predictable relationship between written symbols (graphemes) and sounds (phonemes). This results in relatively simple phonological processes compared to those of fusional languages. According to Bender[12], the complexity of phonological processes can obscure morpheme boundaries, making them less identifiable. This transparency and simplicity limit the number of root morphemes in an agglutinative and orthographically transparent language like Turkish compared to fusional languages. However, at the same time, word realizations tend to be longer, which exacerbates the *orthographic sensitivity* problem we defined. Consequently, Turkish is well-suited for representation through a state machine with fewer node instances, provided the atomic roots are identified and the numerous affixation (transitions) rules of the language are modeled. Such a structure makes it easier to avoid noise in the semantic space while fostering creativity at the subword level, enabling the handling of rare words, OOV words, and even made-up words effectively.

To conclude, the synthesis level and orthographic transparency level of a synthetic language determine the effectiveness of our morphological modeling approach in reducing noise within semantic spaces. In languages closer to the pictographic typology, noise issues are largely absent, eliminating the need for such solutions. These factors form one of the key assumptions underlying our approach to modeling languages.

[12] Essential #22: Languages vary in how easy it is to find the boundaries between morphemes within word (Bender, 2013).

3.3 SIM-REL SPACE

Instead of dividing word-pairs into two separate categories, we opted to assign two scores to each word-pair: one for similarity and one for relatedness. This approach enables us to maintain the dataset as a unified entity while assessing semantic models across both types of relationships. The two-dimensional structure of our evaluation data allows for the visualization of the dataset's semantic space using a scatter plot, which we refer to as the "Sim-Rel vector space" (Figure 3.2).

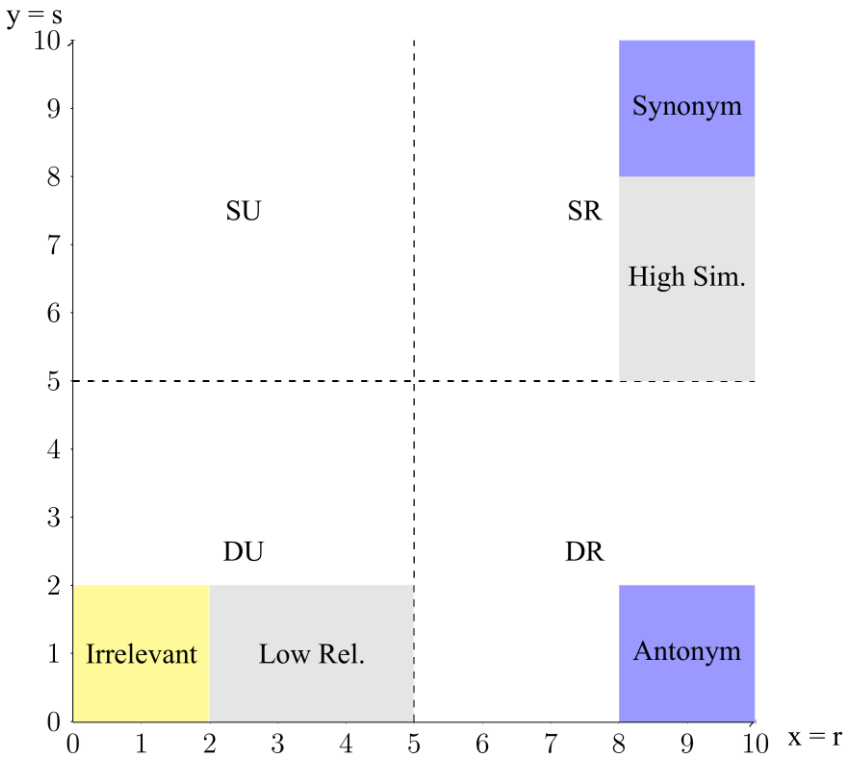

Figure 3.2 Sim-Rel vector space of word-pairs.

Given that the x and y axes represent the relatedness (r) and similarity (s) scores of each word-pair in the dataset, with both r and s (orthogonality of paradigmatic and syntagmatic relations) ranging from 0 to 10, the semantic sub-spaces (ss) can be categorized as:

$$ss = f_1(r, s) = \begin{cases} \text{SU}, & \text{if } s \geq 5 \text{ and } r < 5 \\ \text{SR}, & \text{if } s \geq 5 \text{ and } r \geq 5 \\ \text{DU}, & \text{if } s < 5 \text{ and } r < 5 \\ \text{DR}, & \text{if } s < 5 \text{ and } r \geq 5 \end{cases}$$

Let t = 2 represent a threshold variable indicating the boundary point of relation-type spaces, where synonym, antonym, and irrelevant serve as categorical labels for possible semantic relation types rt. The function rt = f_2(r, s) can be described as:

$$rt = f_2(r, s) = \begin{cases} \text{synonym}, & \text{if } 10 - t \leq s \text{ and } 10 - t \leq r \\ \text{antonym}, & \text{if } 10 - t \leq r \text{ and } s \leq t \\ \text{irrelevant}, & \text{if } t \geq r \text{ and } t \geq s \end{cases}$$

If we assume participants are instructed to assign lower similarity scores s (closer to 0) for antonym judgments on word-pairs and higher scores (closer to 10) for synonym judgments[13], the following points can be inferred based on the Sim-Rel vector space functions f_1 and f_2 described above:

- An ideal DSM would be able to classify word-pairs into every semantic sub-space ss with complete (100%) accuracy.
- No word-pair instance should be classified into the similar-unrelated SU sub-space. Semantically, all highly similar word-pairs are also expected to be highly related. For instance, the word-pair "*car* - *automobile*" is highly similar and is likely to share many common contextual neighbors, which would result in a high relatedness score.

[13] Assigning scores closer to zero for antonyms is a standard practice in similarity datasets (Gerz et al., 2016; Hill et al., 2016).

- Word-pairs could be classified as synonyms if their rt value is identified as *synonym*, depending on the t parameter. This same rule applies to the *irrelevant* value. The threshold value t = 2 was intuitively selected for the Sim-Rel semantic space, and for simplicity in modeling and visualization, t = 2 was kept constant across all axes and relations. Further theoretical or empirical investigation into how to select these threshold values is left for future research.
- *Antonym-DR overlap problem*: No DSM can perfectly assign rt as an antonym. The boundaries between antonyms and dissimilar-related (DR) word-pairs are often semantically blurred. Our bi-dimensional evaluation model is unable to clearly distinguish between them. For example, the word-pairs "*tense - loose*" and "*red - rose*" may receive similar r and s scores, even though the former is an antonym, while the latter clearly is not. In similarity datasets, asking participants to score lower for antonym judgments is a common approach (Hill et al., 2016; Gerz et al., 2016). However, the root of this problem lies in storing two distinct relationship types (synonym, antonym) within the same s variable, which is a limitation of the Sim-Rel vector space model. We leave addressing this issue for future research.

3.4 OSIM-REL SPACE

We define **OSim-Rel** space (Figure 3.3) following the idea of Sim-Rel space from the AnlamVer study by Ercan and Yıldız (2018). The Sim-Rel space is a Cartesian coordinate system where each axis represents scores of specific type of relations (x:relatedness, y:similarity) for same word-pairs. Each word-pair has two distinct scores, allowing them to be represented as a single point in the space (x = rel(w1, w2), y = sim(w1, w2)). This conceptual space enables researchers to categorize word-pairs into sub-regions based on certain assumptions about specific semantic relations within the space. For example, word-pairs can be categorized as synonyms if they have high relatedness and

similarity scores (sim(w1, w2) > 7.5, rel(w1, w2) > 7.5) (e.g., *car – automobile*), or antonyms if their relatedness is high but similarity is low (e.g., *hard – easy*).

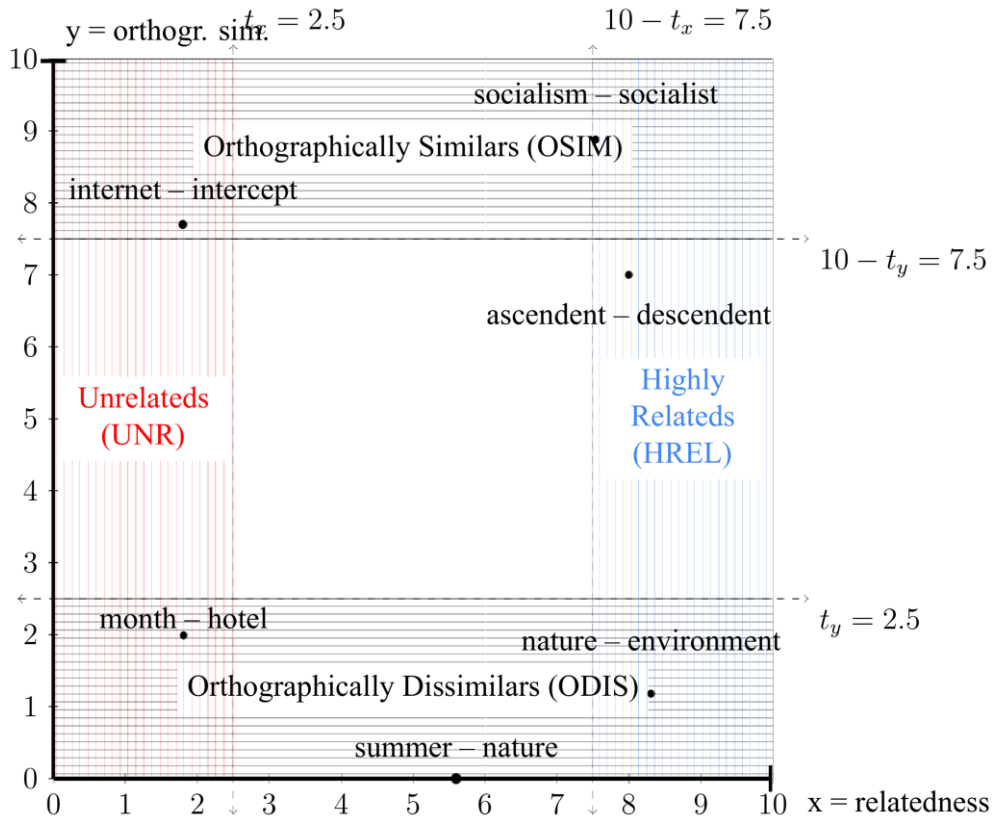

Figure 3.3 OSim-Rel: Orthographic Similarity - Relatedness Space of Word-pairs. Threshold variables t_x, t_y equally selected as 2.5. Unrelateds (UNR) area is vertically hatched in red while Highly Relateds (HREL) area hatched in blue. Orthographically Similars (OSIM) and Orthographically Dissimilars (ODIS) areas in horizontal black lines.

We introduce a modified version of the original Sim-Rel space, representing **orthographic similarity** (OSim) score of word-pairs on y-axis instead of the similarity score. For a given word-pair [w1, w2], we calculate OSim(w1, w2) orthographic similarity scores of two words. While the y-axis can be easily calculated for every possible word-pair, the relatedness values of x-axis (x = Rel(w1, w2)) should be obtained from existing relatedness datasets, DSMs such as FastText, or WordNet-based relatedness/similarity approximation

algorithms, which we cover in §5.3.2. We define t_x and ty ($0 < t_x < 5$, $0 < ty < 5$) as threshold variables that determine decision boundaries for x and y axes respectively. Figure 3.3 illustrates how the conceptual OSim-Rel space defines ss sub-spaces with the function f_1 (Equation 3.1), where t_x and t_y values are equally chosen as 2.5. With this configuration, a word-pair such as *internet – intercept* will reside at the intersection of the orthographically-similars (OSIM) and unrelateds (UNR) sub-spaces since it has a high orthographic similarity score of 7.7/10 and a low relatedness score of 1.8/10. We arbitrarily select t_x and t_y threshold values of 2.5 in order to divide the OSim-Rel space into symmetrical sub-spaces and sub-regions. Therefore, the sub-spaces OSIM, UNR, HREL, and ODIS in Figure 3.3, and the sub-regions such as OSIM-UNR Q3/Q4 in Figure 3.4, illustrate the basic assumptions of this study in determining degrees of orthographic similarity and relatedness measures.

$$ss = f_1(w_1, w_2) = (x = Rel(w_1, w_2), y = OSim(w_1, w_2), t_x, t_y) = \begin{cases} \text{OSIM}, & \text{if } y \geq 10 - t_y \\ \text{ODIS}, & \text{if } y < t_y \\ \text{UNR}, & \text{if } x < t_x \\ \text{HREL}, & \text{if } x \geq 10 - t_x \end{cases} \quad (3.1)$$

$$sr = f_2(x = Rel(w_1, w_2), y = OSim(w_1, w_2), t_x, t_y) = \begin{cases} \text{OSIM-UNR Q4}, & \text{if } y \geq 10 - t_y \text{ and } x < t_x \\ \text{OSIM-UNR Q3}, & \text{if } 10 - t_y \geq y \geq 10 - (2 \times t_y) \text{ and } x < t_x \\ \text{OSIM-HREL}, & \text{if } y \geq 10 - t_y \text{ and } x \geq 10 - t_x \\ \text{ODIS-UNR}, & \text{if } y < t_y \text{ and } x < t_x \\ \text{ODIS-HREL}, & \text{if } y < t_y \text{ and } x \geq 10 - t_x \end{cases} \quad (3.2)$$

Since every word-pair should reside on two sub-spaces in OSim-Rel, we define a second function f_2 to label given word-pairs into single sub-regions (Equation 3.2). As Figure 3.4 highlights in yellow, orthographically-similar-but-unrelated (OSIM-UNR) **Q3 and Q4 sub-regions are the main focus** of this thesis. We add the Q3 sub-region into our dataset to have more word-pairs (more than 99% of all word-pairs, Table 5.11) and to be able to measure the contribution of orthographic similarity to performance (see left-to-right trend in

Figure 6.3). Figure 3.4 also shows how the average scores of conventional wordsim datasets (blue and red points) are far from addressing the Q3 and Q4 OSimUnr cases.

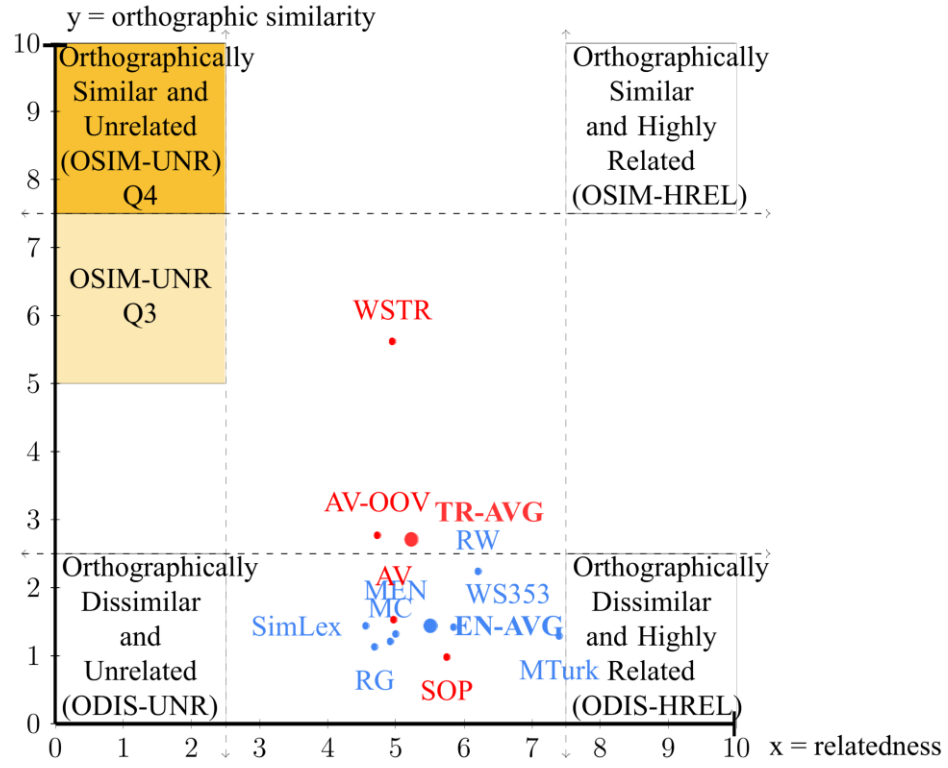

Figure 3.4 Sub-regions of OSIM-REL Space. Points represents average scores of wordsim datasets (RW: Rarewords, SOP: Sopaoglu, AV: AnlamVer, WSTR: WordSimTR). Bold points denote average wordsim score for each language (EN-AVG, TR-AVG). Red and blue dots denote Turkish and English datasets. Area in yellows (OSIM-UNR) are the main focus of this study. All dataset scores are normalized to [0-10] scale.

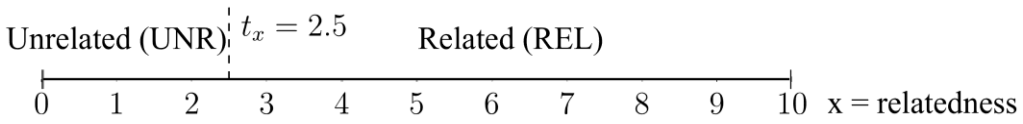

Figure 3.5 Assumptions on the relatedness axis of word-pair scoring

3.4.1 Selecting Orthographic Similarity Algorithms

The first orthographic similarity measure we utilize is the *edit distance* algorithm, which is easy to implement and computationally efficient for word-level lengths. It is particularly well-suited for modeling spelling mistakes, as it calculates the number of *edits* required to transform one text into another. To convert the normalized version of the edit distance algorithm from a distance measure to a *similarity* measure, we apply the formulation in Equation 3.3.

$$editsim = y = OSim(w_1, w_2) = 1 - NormalizedEditDistance(w_1, w_2) \quad (3.3)$$

We refer to the inverted version as edit *similarity* or editsim. While editsim is useful for benchmarking, it may not be the best fit for our specific needs due to its **four significant downsides**. Firstly, since it operates at the character level, it may not always align with human orthographic similarity intuition and may not adequately model morpheme overlaps. Since the insert/delete/modify edits can occur at any word index, a few modifications can entirely change a word to something else. For example, the word-pair *aerobics – heroin* receives an editsim score of 0.5, even though the words don't share any morphemes (Table 3.4). Secondly, editsim yields low scores when the lengths of the two words differ significantly. For instance, the word-pair *göz (eye) – gözlükçülük (occupation of being an optician)* receives an editsim score of 0.27, despite the two words sharing the same root *_göz*. We want our dataset to include word-pairs that are different in length and possibly share some morpheme-like blocks. Thirdly, editsim tends to yield higher scores than we expect for very short word-pairs, as in the example *car – bar* (0.67). Lastly, similar to the third point, when the edit differences are at the beginning of a word, editsim still yields very high scores for word-pairs with completely different roots, such as *legging – begging* (0.74). In such cases, it does not pose a significant challenge for models to distinguish words with completely different meanings. The OSimUnr dataset will include orthographically-similar word-pairs with scores higher than 0.5.

Therefore, we aim for orthographic similarity algorithms to yield higher scores for word-pairs that are most likely to have **morpheme-like block overlaps**, rather than character-level distances. To address these issues, we conducted experiments with various orthographic similarity algorithm configurations, as shown in Table 3.4, in search of alternatives that meet our study requirements. Our goal is to compare and correlate orthographic similarity scores with our normalized model predictions. As a result, we exclude non-normalized candidates, such as q-gram and longest common subsequence (LCS) algorithms (Hirschberg, 1975), from consideration.

Table 3.4 Comparison of normalized orthographic similarity algorithms. Selected algorithm configurations editsim and over_ft23 are displayed in bold.

Word-pair	**edit sim**	ngr2	ngr3	dice 2gr	over 3gr	over ft[1-3]	over ft(2-3)	**over ft[2-3]**	jacc ft[2-3]	dice ft[2-3]	over ft[2-6]	over ft[3-6]
car – bar	0.67	0.50	0.39	0.50	0.00	0.58	0.33	0.43	0.27	0.43	0.30	0.17
verbaliser – verbalizer	0.90	0.90	0.90	0.75	0.62	0.80	0.69	0.75	0.60	0.75	0.61	0.56
aerobics – heroin	0.50	0.44	0.40	0.33	0.25	0.43	0.33	0.23	0.11	0.20	0.12	0.06
natural – contrary	0.25	0.25	0.21	0.15	0.00	0.26	0.09	0.07	0.03	0.06	0.03	0.00
göz – gözlükçülük	0.27	0.36	0.36	0.40	1.00	0.83	1.00	0.71	0.23	0.37	0.60	0.50
legging – begging	0.86	0.79	0.74	0.83	0.80	0.77	0.82	0.73	0.58	0.73	0.67	0.64
condor – condom	0.83	0.92	0.94	0.80	0.75	0.75	0.78	0.69	0.53	0.69	0.60	0.56
converse – conserve	0.75	0.75	0.79	0.71	0.17	0.69	0.46	0.53	0.36	0.53	0.29	0.12

Among the candidates, n-gram similarity (i.e., ngr2 or ngr3) stands out as it measures above-character-level similarities in a recursive fashion. Notably, according to (Kondrak, 2005), the longest common subsequence and editsim algorithms are special cases of n-gram similarity. While n-gram similarity performs slightly better than editsim for the first, third, and fourth problems, it still yields low scores, such as 0.36 for the word-pair *göz – gözlükçülük*, when addressing the second problem. We aim to include more challenging word-pairs with varying word lengths, emphasizing shared morpheme-like structures (e.g., *communicant – commute*), which are difficult for semantic models to distinguish. This becomes especially crucial when the models' objectives are simple and

sensitive to overlapping segments, as in the case of the morphologically segmented models of this study, FT-M and FT-MR.

To maintain the n-gramming (i.e., *shingles* in this context) based comparison of the algorithm, we utilize FastText's default n-gramming algorithm (Table 3.1), which places higher value on the beginning n-grams by adding beginning characters ('<') to words before generating n-grams.[14] Compared to a fixed-length n-gram algorithm, FastText's n-gramming offers greater flexibility in representing morphemes consisting of two, three, or four characters. This flexibility is achieved by generating n-grams of varying lengths. Based on our observations presented in Table 3.4, we select its *ft[2-3]* configuration, which combines 2-grams and 3-grams, as it better models morpheme similarity. This choice appears reasonable considering that the average character size of the top 100 most frequent suffixes in our English corpora is 2.7 (2.82 for Turkish), which falls between 2 and 3. Finally, the *overlap coefficient*[15] is employed to address the third problem length-mismatch, by dividing the number of overlapping segments (i.e., seg) by the minimum number of elements in the two sets (Equation 3.4). This coefficient provides a measure of similarity that accounts for overlapping segments between words.

$$overlap(seg_{w1}, seg_{w2}) = \frac{|seg_{w1} \cap seg_{w2}|}{min(|seg_{w1}|, |seg_{w2}|)} \tag{3.4}$$

The overlap coefficient (overft* columns in Table 3.4) is unique among segment-comparing coefficients because it normalizes the difference in the number of segments being compared. This is in contrast to other similar coefficients such as Jaccard and Dice, as illustrated in the last columns of the *göz – gözlükçülük* (jacc=0.23, dice=0.37, over=0.71) row in Table 3.4.

[14] Square brackets in "ft[2-3]" indicate beginning and ending characters are included in the n-grams.
[15] Also known as the Szymkiewicz – Simpson coefficient.

Consequently, as an alternative orthographic similarity measure, we propose over_ft23, which **combines FastText's n-gramming technique with the overlap coefficient** to select word-pairs that present greater challenges for semantic models to distinguish. To address any potential criticism that the selection of FastText's own n-gramming algorithm might be a biased attempt towards highlighting FastText's n-gram-caused problems, we include the editsim algorithm as a secondary orthographic similarity measure in the thesis. By using editsim alongside over_ft23, we ensure a fair and comprehensive evaluation of the word-pairs, allowing us to explore the distinguishing ability of semantic models in different scenarios. This approach helps us **avoid any potential bias** and provides a more robust analysis of model performances. We should note that our experiments show that FastText's char-gram segmentation fails to identify unrelated word-pairs that are generated by both measures (editsim: below 5.82, over_ft23: below 4.18), while morphological segmentation outperforms it by a substantial margin (best:70.94 worst:64.82, Table 6.4). As anticipated, the final editsim sub-dataset contains more word-pairs (≈570K) than the final over_ft23 sub-dataset (≈70K), as shown in Table 5.11. Our experiments demonstrate that the over_ft23 dataset poses greater challenges for semantic models, as evidenced by the lower accuracy of our best performing model, MR on, over_ft23 (64.82%) compared to editsim (68.47%). For most algorithm implementations, we utilize the python-string-similarity package.[16] To enhance runtime performance, we cythonize (Behnel et al., 2011) the library, meaning that converting it to its C programming-language equivalents.

[16] https://github.com/luozhouyang/python-string-similarity

CHAPTER 4

4. TURKISH MORPHOLOGY

4.1 LANGUAGE STRUCTURE

Turkish is an agglutinative language with productive inflectional and derivation morphology. Agglutinative nature makes it possible to form new words by stringing stem, morphemes and suffixes together. Due to the Turkish is bound-morpheme, there can be only one lexical stem (root) of a word exists. Inflections can be formed in two ways:

- Inflectional suffixes: Suffixes can form a stem to express new functions or attributes such as: mood, person, tense. For example, the word *gözleri* (their eyes) can be expressed by suffixes like *_göz+ler+i*, more formally: "göz+NOUN+A3PL+PNON+ACC"
- Derivational suffixes: Suffixes that form a new word while possibly changing the the syntactic structure of the word. For example, the word *gözsel* (about eye) can be expressed by suffixes like *göz+sel*, formally: "göz+NOUN+A3SG+PNON+NOMDˆ B+ADJ+RELATED"

Turkish has a free constituent order. For instance, all possible word combinations of the sentence "yaşa, hayatını, aşkla" (live your life with love) would be grammatically and semantically correct: "hayatını aşkla yaşa" (live your life with love), "aşkla yaşa hayatını" (live your life with love), ...

4.2 MORPHOLOGICAL DISAMBIGUATION

Morphological disambiguation is the problem of selecting the sequence of morphological parses (including the root). For instance, the noun phrase *evin terası* (the terrace of the house) have the following parses below:

Table 4.1 Disambiguation example taken from (Hakkani-Tür et al., 2000). Correct parses denoted in bold.

	evin	terası
1	evin+Noun+A3sg+Pnon+Nom	**teras+Noun+A3sg+P3sg+Nom**
2	ev+Noun+A3sg+P2sg+Nom	teras+Noun+A3sg+Pnon+Acc
3	**ev+Noun+A3sg+Pnon+Gen**	

4.3 COVERAGE STATISTICS

As being a highly productive inflectional language, a single Turkish lexicon (root word) can take thousands of surface forms (Sproat, 1992) which means there is a strong possibility that a trained model hasn't seen a highly inflected testing word in the training set, so-called out-of-vocabulary (OOV) problem. Turkish has a high morpheme/word type ratio (more than 3) (Jurafsky, 2000; Sak et al., 2012; Oflazer, 1996) where most of morpheme transitions only *slightly changes* the root meaning (not derivational).

Our own corpus coverage analysis (news dataset) shows that 47% of word types (277K) occur only once in the corpus which is compatible with Sak's report on Boun Corpus coverage statistics (Sak et al., 2011). Removing all the least frequent word types less than or equal to 10, would be cutting off 84% of the word types. Table 4.2 displays frequency-of-frequency of word types (unique token) based on coverage analysis on our own corpus (580K word types, 16M words in total). Given a DSM implementation cut off least frequent words less than 10 as being rare-words, for Turkish 84% words would be rare-words. Moreover, OOV rate of Turkish is so much higher than a less inflective language.

Table 4.2 Word type frequency analysis of our corpus consists of 580K word types in total (vocabulary size). Fof denotes frequency of frequencies, Cumm. denotes cumulative Fof.

Freq.	WordTypes (Fof)	WordTypes (Cumm.)	WordTypes (%)
1	277,928	277,928	47.86
2	79,711	357,639	61.59
3	40,677	398,316	68.60
4	25,732	424,048	73.03
5	18,247	442,295	76.17
6	13,542	455,837	78.50
7	10,655	466,492	80.34
8	8,626	475,118	81.82
9	7,267	482,385	83.08
10	6,158	488,543	84.14

4.4 CORPORA

As in many unsupervised representation learning tasks, DSMs require huge datasets. One of the biggest publicly available Turkish corpus (Boun Corpus) is about 0,42B words (Sak et al., 2011). Most DSM experiments reported increasing embedding quality with the increasing corpus size (see Table 4.3). Mikolov et al reported 6% increase in model quality by increasing corpus size from 6B to 33B (Mikolov et al., 2013b). Considering corpus sizes of experiments ran by DSM researchers vary from 1.5B to 33B, we will need more textual unannotated data to train. We set out target to compile corpora consisting of 5B tokens, collected from various sources, including existing corpuses (Boun Corpus, Wikipedia TR etc.).

Table 4.3 Corpus sizes of various DSM experiments.

#	Corpus	Author	Lang.	Words#
1	WikiEn2013	Levy et al.	EN	1.5B
2	EnWiki2016	Fares et al.	EN	2B
3	Gigaword5thEd	Fares et al.	EN	4.8B
4	GoogleNews	Mikolov et al.	EN	33B
5	GoogleNews	Mikolov et al.	EN	6B
6	GoogleNews (subset)	Mikolov et al.	EN	783M
7	No name	Levy et al.	EN	10.5B
8	Wiki,WaC,British Nat.	Baroni et al.	EN	2.8B
9	wiki2010	Qiu et al.	EN	1B

4.5 ASSUMPTIONS ON DERIVATIONAL MORPHOLOGY

In our investigation of the role of prior morphological knowledge in subword-level modeling and evaluation, we believe that the root cause of the *overlapping-n-grams* and *orthographic-sensitivity* problems lies in the lack of knowledge in identifying the appropriate sub-units that represent the meaning of words. To address these issues, we make certain assumptions regarding language and morphology. Throughout the study, we follow the *-prefix1... -prefixp_root1... _rootr+suffix1+... +suffixs* format for morphological segmentations (e.g., *-co_here+ance+y* for coherency).

4.5.1 The Meaning is on the Root(s)

Morphological segmentation is a process that divides words into their constituent morphemes, which are the smallest meaningful units of language. Morphemes can be further categorized into roots and affixes (prefixes or suffixes). Every word contains at least one root (i.e., stem) morpheme. **Root morphemes convey core lexical meanings** of words (Bender, 2013, Essential #11).[17] English is a fusional language; therefore, it supports compounding of words, which can form multiple root morphemes per word (e.g., *_dog_house* for

[17] Essential #11: Root morphemes convey core lexical meaning (Bender, 2013).

doghouse). In Turkish, although most compounds are written as separate words (e.g., *kız arkadaş* for *girlfriend*), it is worth noting that Turkish words can have multiple roots in practice, as seen in the example *oniki* (twelve), formed by combining the words *on* (ten) and *iki* (two).

4.5.2 Words Derived from the Same Root are Related

Whether a derivation is compositional (e.g., *_age+less*) or non-compositional (e.g., *_butter_fly*), the derived words slightly change the meaning of the root word. We assume that such derived words have a syntagmatic relation with the root word, meaning that they tend to occur in similar contexts (e.g., *_symbol – _symbol+ism*). This assumption also applies to words that result from different suffixations sharing the same root (e.g., *_theor+y – _theor+ist*), as well as to words with multiple levels of derivation (e.g., *_theor+y – _theor+etic+al+ly*). Although derivations can sometimes exhibit idiosyncratic patterns, if two words are derived from the same root, we consider them to be related.

4.5.3 Compound Words are Related to their Constituents

We assume that if a word is a compound, it is inherently related to its constituents, regardless of whether the composition is idiosyncratic or regular. For instance, the compositional compound *doghouse* is related to *dog* and *house* to some extent. Similarly, *butterfly* is related to *butter* and *fly* even though the original meanings of the individual words may have evolved or become less transparent over time.

4.5.4 Derivational Affixes Change the Meaning

The core meaning of a word is attributed to root morphemes, which serve as a foundation for deriving new words with distinct meanings through the process of derivational suffixation (by prefixes or suffixes), as exemplified by

the word *_king+dom*.[18] Additionally, derivational processes can also alter the part-of-speech of a word, as seen in the example *_compose+it+ion* (V→N). Both the Turkish and English languages have a diverse inventory of derivational affixes (Bender, 2013).

4.5.5 Inflectional Affixes do not Change the Meaning

Unlike derivational suffixes, inflectional affixes do not alter the meaning of root words. Instead, they primarily contribute important semantic or syntactic features[19], such as tenses (e.g., *_run+s*), aspects (e.g., *_do+ing*), or plurality (e.g., *_table+s*) at the sentence level. In contrast, in the word-level context, inflections do not fundamentally change the meanings of words. Turkish, as an agglutinative language, exhibits extensive inflectional patterns, while English has more limited use of inflections.

4.6 MODELING DERIVATIONAL MORPHOLOGY

We utilize morphological information for two distinct purposes: a) to facilitate automatic dataset generation by detecting shared roots, and b) to model atomic sub-units of language for training.

4.6.1 Root Detection

While a comprehensive morphological analysis is essential for modeling, for dataset generation, a *DetectRoots()* root detection implementation is sufficient. The function returns the morphological root or roots of each given word. During the automatic dataset generation phase, the primary objective of morphology is to answer the query *IsRelated()* for given word-pairs. Based on the assumption that ”words derived from the same root are related” (§4.5.2), we consider two words to be related if we identify that they share at least one of

[18] Essential #12: Derivational affixes can change the lexical meaning (Bender, 2013). Example from the book.
[19] Essential #14: Inflectional affixes add syntactically or semantically relevant features (Bender, 2013).

their roots. For instance, when we identify that the word-pair *criminal – decriminalization* both originate from the root *_crime*, we can confidently conclude that they are related without requiring a precise degree of their relatedness.

4.6.2 Atomic Roots

When referring to *root* words, unlike in many NLP studies, our goals require going beyond the mere removal of simple derivations and inflections. We decompose the words into their **most fundamental atomic root forms**, sometimes necessitating tracing the words back to their historical origins. For instance, based on the MorphoLex database (Sánchez-Gutiérrez et al., 2018), the words adhere, *inherent*, and *coherence* share the same root *_here*. However, they do not share the same root with inherit or nowhere, which have the roots *_herit* and *_where*, respectively. Due to the dynamic nature of language, words and morphemes have undergone fusion, change, and borrowing from other languages over time. As published by the MorphoLex database, the word *nevertheless* can be analyzed as "{(never)}{(theo)}{(less)}"[20] even though its current meaning may have shifted. This analysis is based on its root *theo*, arguably originated from the Greek word *theos* (meaning 'the god'). Such analysis requires a separate field of study that encompasses linguists and historians. If the arguable groundtruth root of the word nevertheless were not *_theo*, we would incorrectly (false positive) filter out the word-pair *nevertheless – atheism* from the dataset because we assumed that they share the same root.

4.6.3 English Stack

As an initial step in our English morphology stack, we employ the Morphy, a built-in lemmatizer tool provided by the NLTK framework (Bird et al., 2009). This rule-based library can handle commonly used suffix inflections (but not prefixes), such as *+ing*, *+s*, and *+ed*, to separate basic inflections and identify

[20] This is MorphoLex's syntax for morphological decompositions.

simple roots. In the second step, we utilize the MorphoLex database, which contains static analyses for 68,616 surface words. We parse the recursive syntax of MorphoLex (e.g., ”{(psycho)(log)ic»al>}>ly>”) and convert it to our representation of morpheme sequences. To maintain consistency in handling **allomorphic realizations**, MorphoLex utilizes meta affixes such as ”>ize>” to represent different variations of morphemes such as *iza, ize, isa, ise*. Similarly, the meta affix ">able>" represents morphemes found in words like *accept<u>abil</u>ity* and *accept<u>able</u>*. While having meta morphemes can be advantageous, generating the same meta affixes is not always possible, especially in cases where words are not included in MorphoLex's vocabulary. Within MorphoLex, similar to meta affixations, there exists meta root forms that differ from their surface realizations. For example, the meta root form ”(crimin)” fully represents the surface word *crime*, while the meta root ”(theo)” serves as the root of the surface word *atheist* (”<a<(theo)>ist>”). While detecting the roots alone is sufficient for generating the dataset and for our root-only model FT-MR, our fully morphological model FT-M requires us to utilize MorphoLex expressions (with roots and affixations) as the primary source of morphological analysis for English.

Morfessor2 (Virpioja et al., 2013) is a supervised model trained using the Conditional Random Field method. While it offers consistent string segmentation, it lacks a morphological knowledge base and does not align with our meta roots and affixes. As a result, we chose not to include it in our stack. As shown in our benchmark (Table 4.4), the Morfessor2 model exhibits incorrect root predictions (*_activ*, *_char*, *_bodi*), especially in cases involving proper nouns like country and language names. This issue is likely attributable to the absence of a lexicon-based approach. We use the Morfessor2 implementation through the Polyglot library (Al-Rfou et al., 2013). Another method we employ utilizes the *derivationally-related-form* association of lemmas from WordNet (WN-DR column in Table 4.4). Although this method is not a morphological approach per se, it allows us to leverage the knowledge pool of shared root

relationships. Therefore, we included the *derivationally-related-form* information in our filtering pipeline (§5.3.2.4), rather than the morphology stack.

4.6.4 Stacking and Shallow Affixation

MorphoLex offers precise analyses that align well with our requirements, but its vocabulary is constrained. Specifically, it faces difficulties in handling loan words, domain-specific terminologies, and compounds. Instead of expanding its vocabulary manually, we employ a combination of resources, including Morphy, WordNet, and our pool of affixes. Through a simple suffixation algorithm, we apply these resources to **convert MorphoLex from a mere lookup table into a shallow morphological analyzer** tailored for English.

Table 4.4 Hand-picked examples from Shared Root Detection experiments for English. Symbol ×denotes that the task is failed detecting shared root. The "ok [root]" pattern denotes that the task passes detecting shared root 'root'. The "ok" cells of WN-DR denote that WordNet has prior knowledge that two words are derivationally-related without knowing the actual roots. *The Full Stack combines Morphy, MorphoLex, and WordNet (with its word-pool only) with shallow affixations. Morfessor2 is not included in the English stack.

Word-pair	Morphy	WN-DR	Morfessor2	MorphoLex	Full Stack*
activism – activist	×	ok	×[activ]	ok [act]	ok [act]
atheist – theist	×	×	×	ok [theo]	ok [theo]
athene – athens	×	×	×	×	×
bucharest – bucharesti	×	×	×[char]	×	ok [bucharest]
cambodia – cambodian	×	ok	×[bodi]	ok [cambodia]	ok [cambodia]
dog – dogs	ok [dog]	x	ok [dog]	ok [dog]	ok [dog]
psychophysics – physics	×	×	ok [physics]	×	ok [physics]

Firstly, we create a comprehensive candidate word pool by combining WordNet lemmas with the surface and root forms from MorphoLex. WordNet is powerful at domain-specific words (e.g., *byra* [a genus of a flowering plant]) and proper nouns (e.g., *Aristotelia*, *Google*). For words that do not yield a root from MorphoLex, we apply *shallow* affixation after stripping off their inflections with Morphy. We use the term *shallow* because we do not represent morphemes with a hierarchical structure as we do in Turkish morphological analysis. Instead,

it is a simple rule-based string manipulation. It involves conducting trials with prefixes and suffixes for each surface word query, limited to the extent of the available affixes. We check if these trials match with a word or a root from our candidate word pool. For example, although *cinematograph* has the analysis of "{(cinema)}>tograph>", *cinematographer* is not present in MorphoLex. By removing the candidate meta suffixes (e.g., *+er*) from the query, we check if the remaining result matches a root or a word in our pool. This approach allows us to obtain multiple shallow analyses such as *_cinema+tograph+er*. Similarly, assuming the given query might be a compound word structure, we concatenate it to our available roots, enabling us to analyze words like psychophysics (*_psycho+physic*), that are not available in our database.

The stacking operations we employ allow us to **augment** our available morphological analyses with a complexity of O(R + S + A) for each surface word query, where each letter represents number of items for that type (R: roots, S : surfaces, A: affixes). Since this task focuses on word-pair-based queries, it does not require contextual information beyond individual words. As a result, there is no need for a sentence-level or higher-level disambiguation agent. Due to the word-based nature of each analysis, we can easily create word-analysis cache tables to optimize runtime performance. As each surface word is analyzed only once, the overall computation complexity for all possible queries becomes O(QueryW ords × (R + S + A)).

4.6.5 Turkish Morphological Analysis

Modeling morphology solely based on static analyses using tools such as MorphoLex, is not feasible due to the rich inflectional nature of the Turkish language. Turkish words can have an infinite number of surface forms, as exemplified by a word like *pencerelerimizden*, which derives from the root *_pencere* (window) through various inflections.

Drawing on the principles of two-level morphology (Koskenniemi, 1983), analyzers typically aim to transform *surface representations* into *underlying representations* (lexicons) using rewrite rules that govern productive

derivations and inflections within a language. A study by Yıldız et al. (2019) provides a comparison of various morphological analyzers (including the one we extend) documented in the existing literature for Turkish. However, none of the analyzers in the literature provides the level of detail in lexicons and derivational suffixation structure required to reduce to atomic roots, which aligns with the objectives of our work. The lexicons of general-purpose morphological analyzers often contain many already derived words (e.g., *gözlükçülük* or *gözlemcilik*) because they borrow the words from meaning databases like WordNet or national dictionaries. In contrast, our goal is to model derivations down to the most atomic roots.

For the purpose of customization, we extend Turkish Morphological Analysis Java library (Yıldız et al., 2019)[21], utilizing its lexicon and meta rule engine for suffixation executed by its built-in finite state transducer. While its original file turkish_finite_state_machine.xml has 1,565 rules for state transitions, we expanded it to 1,821 rules. Notably, we added various meta suffixes like *+loji (anjiyoloji)*[angiology], *+grafi (anjiyografi)*[angiography], *+ör* (*anket+ör*) [pollster] to facilitate the derivation of foreign-origin words and affixes. Table 4.5 shows sample lexicon and suffixation rule definitions from our implementation.

[21] https://github.com/olcaytaner/TurkishMorphologicalAnalysis

Table 4.5 Sample definitions from TurkishMorphologicalAnalysis library customization. Customized lexicon includes 62,575 entries. Customized suffixation engine includes 1,821 transition rules (with blocks). CL_ISIM: Noun, IS_OA: Proper noun, FRG: Foreign derivation, ˆDB: Derivation.Hand-picked examples from Shared Root Detection experiments for English.

Lexicon (txt file)

```
..
anjiyo CL_ISIM
anket CL_ISIM
anketör CL_ISIM FRG
yön CL_ISIM
yönetim CL_ISIM IS_OA
yönetme CL_ISIM
yönetmelik IS_SD CL_ISIM
kıpkırmızı IS_ADJ
kitapsever CL_ISIM
göz CL_ISIM
gözleme CL_ISIM ATOM
gözlemeci CL_ISIM
gözlemecilik CL_ISIM IS_SD
gözlemcilik CL_ISIM IS_SD
gözlük CL_ISIM IS_SD
gözlükçü CL_ISIM
gözlükçülük CL_ISIM IS_SD
..
```

Suffixation Rules (xml file)

```
<state name="NominalRoot" start="yes" end="no" originalpos="NOUN">
    <to name="NominalRoot">
        <with name="^DB+NOUN+At" topos="NOUN">At</with>
        <with name="^DB+NOUN+GRAPHY" topos="NOUN" der="1">grafi</with>
        <with name="^DB+NOUN+LOGY" topos="NOUN" der="1">loji</with>
        <with name="^DB+NOUN+FRG-EUR" topos="NOUN" frg="1">ör</with>
    </to>
</state>
<state name="VerbalStem" start="no" end="no">
    <to name="NominalRoot">
        <with name="^DB+NOUN+INF2" topos="NOUN" der="1">mA</with>
    </to>
</state>
<state name="Case1" start="no" end="no">
    <to name="Nominative"><with>0</with></to>
    <to name="Adjective">
        <with name="^DB+ADJ+FITFOR" topos="ADJ" der="1">lHk</with>
    </to>
</state>
```

4.6.6 Turkish Atomic Disambiguation

During the analysis stage, as the number of affixation rules increases, the generation of candidate analyses for a surface form also increases, posing a specific problem in terms of disambiguation. To tackle this, instead of relying on a sentence-level disambiguator, we build a word-level, rule-based disambiguator. This atomic disambiguator utilizes a scoring system based on rules that prioritize the shortest and most frequently occurring morphemes whenever possible. As lexicons can include both roots and affixes that may overlap with each other (e.g., *yönetme* ⊃ *yön*, *oloji* ⊃ *loji*), this disambiguator focuses on selecting the most atomic roots feasible, expecting semantic models to reconstruct derivations in modeling phases. For example, consider the word *yönetmelik* (regulation), which is present in the lexicon as a noun (*CL_ISIM*). The lexicon also contains the related words *yönetme* (management), *yönet* (manage), and *yön* (direction), all of which share the same root. Consequently, as illustrated in Figure 4.1, it generates multiple parse alternatives that include

these words. By utilizing **a scoring system designed to identify atomic morphemes**, the disambiguation process selects the word analysis with the highest score. Upon examining the selected analysis *_yön+At+mA+lHk*, it is observed that it aligns with the static analysis provided by Turkish Morpholex (Arıcan et al., 2022). However, it should be noted that this alignment is not always the case, and when a static analysis is available, it is preferred.

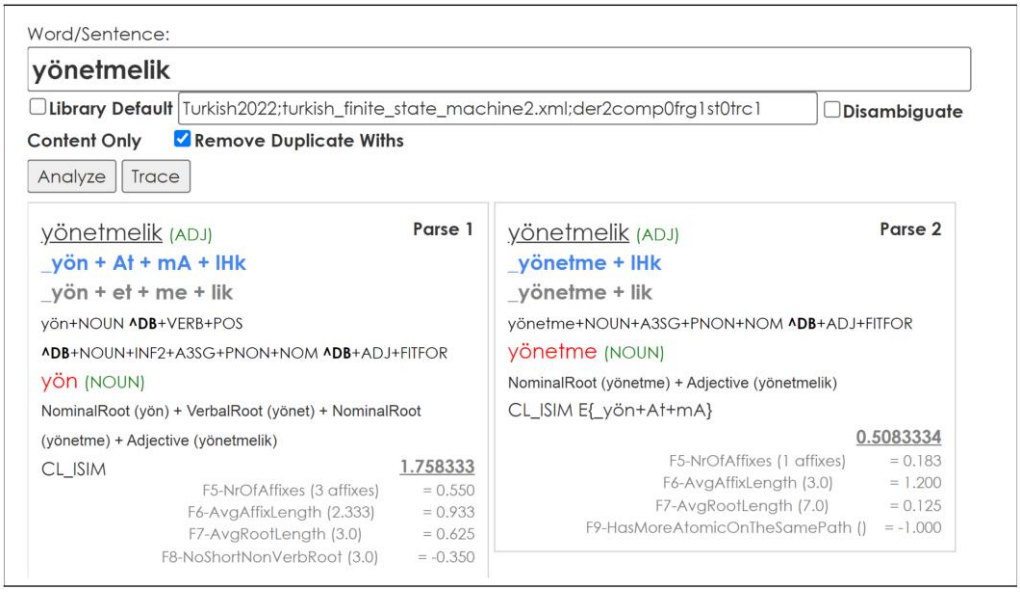

Figure 4.1 Example of an atomic morphological analysis with disambiguation scores. The screenshot is taken from our morphological analysis and disambiguation user interface implementation.

In addition to segmentation, the morphological analyzer offers more information. Examining the same example, it reveals the state changes calculated by the finite state transducer (i.e., FST) along with the corresponding morphological tags: ”yön+NOUN ˆDB+VERB+POS ˆDB+NOUN+**INF2** +A3SG+PNON+NOM ˆDB+ADJ+FITFOR”. While the last derivation *+lHk* is correct as a meta suffix form, there is a debatable transition *FITFOR*, converting the word into an adjective. In some cases, without context, it becomes challenging to determine whether a word should be classified as an adjective or a noun. In this particular case, lacking context, it would have been more accurate

for the word *yönetmelik* to conclude with the *+lHk* suffix as a noun instead of an adjective. Similarly, for the word *yönetme* (management), the analyzer produces the same meta form with *mA*, but this time with the *NEG* and *IMP* tags, which convey the negative imperative meaning (don't manage). In the previous example, *mA* was an infinitive form (INF2). "yön+NOUN ^DB+VERB+POS ^DB+VERB+**NEG**+**IMP**+A2SG". Since we don't have such morphological tags in our English segmentations, to ensure a fair segmentation comparison, this study does not consider the tags and POS information obtained during derivations, such as NEG, FITFOR, IMP. We acknowledge that a simple model like CBOW, used in this study, is not capable of modeling these intricate affixation rules. However, it should be noted that the evaluators employed in this study, such as the relatedness classifier and wordsim, **do not assess compositionality**, which involves language derivation rules. This presents an additional challenge that can be explored in future research. For example, when segmenting the word *yönetmelik* (regulation) as *_yön+At+mA+lHk*, the **valuable original meaning** is lost, making it exceedingly **difficult to reconstruct** the intended meaning from the atomic root *yön* (direction) and the appended suffixes. It is important to mention that FastText also maintains vector representations for surface forms in addition to n-grams.

To prevent the disambiguator from incorrectly segmenting a genuine word from the lexicon into another root, we use a flag called ATOM. This flag indicates that, although the word may have a root, it has either lost its original meaning or its affixation is purely based on phonetic similarity. For example, in the case of *_gözleme+CH* (the one who sells gözleme), although the surface form is derived from the root *göz* (eye), it is more likely related to *gözleme*, a traditional food, with no direct connection to the root (see the example in Table 4.5). By assigning an ATOM flag to *gözleme* in the lexicon, we ensure that the disambiguator assigns a higher score to this root, thus **preventing excessive segmentation**. The use of the ATOM flag helps mitigate over-segmentation by guiding the disambiguator to prioritize the correct interpretation, even when a word shares a root with another but has a different semantic context.

The overall morphological analysis and disambiguation performed for Turkish in this study are comprehensive, extending beyond the scope of this study. The tasks of root detection, morphological analysis, disambiguation, and shallow affixation in this study were performed to the best of our abilities. Instead of solely relying on databases like Turkish Morpholex as a ground truth benchmark to assess the accuracy of our morphological segmentations, our objective was to construct a comprehensive word and affix pool by leveraging all available resources. The systematic evaluation of these tasks and their comparison with the existing literature is deferred to future studies.

4.6.7 Turkish Stack

To compensate for the morphological analyzer's lack of support for compound words and prefixes, we address this issue in the stack stage. Similar to English, we include the Morpholex Turkish dataset (Arıcan et al., 2022) into our stack to improve the overall analysis accuracy. Although the Morpholex Turkish dataset contains a limited number of analyzed words (26,209), its contribution is invaluable in terms of supporting prefixes and compounds. By utilizing the meta roots and prefixes from Morpholex Turkish, we provide support for prefix and compound words through shallow affixation, similar to what we do in English. To enable static analyses from the Morpholex Turkish available for all inflectional surface forms, we incorporate the static analyses from Morpholex Turkish into our analyzer as an additional feature. This integration **combines an extensive inflectional morphological analyzer with the valuable derivational linguistic data**. For example, for the word *kıpkırmızımsı* (crimson reddish), which includes a prefix and is not found in any dataset in its surface form, we can now provide the analysis *-kıp_kırmızı+HmsH*. Similarly, for the compound word *kitapseverlerdendir* (she is one of the booklovers), we can generate the analysis *_kitap_sev+Ar+lAr+DAn+DHr*, while the first three morphemes *_kitap_sev+Ar* (book lover) come from the static compound analysis found in the Turkish Morpholex database. Although Morpholex Turkish is a manually crafted database, since it uses the same meta

suffixes (e.g., *lAr, DAn, HmsH*) as the Turkish Morphological Analysis library, static and dynamic analyses are easily combined.

CHAPTER 5

5. DATASET CONSTRUCTION

5.1 MORPHOLEX TURKISH

In applying a fully derivational methodology, we required a reliable data source that includes words, roots, and affixes in their fully derived forms to serve as the ground truth for morphological analysis and disambiguation. Building upon the original MorphoLex dataset (Sánchez-Gutiérrez et al., 2018), our research group (Arıcan et al., 2022) developed the first comprehensive Turkish morphological lexicon, which contains manually segmented morphological data for 48,472 words. Based on this thesis' morphological assumptions that "the meaning is on the root(s)" and "words derived from the same root are related," this resource also functions as a valuable classifier for word relatedness.

Turkish is an agglutinative language with productive inflectional and derivation morphology. Agglutinative nature makes it possible to form new words by stringing stem, morphemes and suffixes together. Due to the Turkish is bound-morpheme, there can be only one lexical stem (root) of a word exists. Inflections can be formed in two ways:

Table 5.1 Examples of suffixes.

Word	Definition	Root	Suf1	Suf2	Suf3	Suf4	Suf5	Suf6
ölümsüzleştirilme	to be immortalized	öl	yHm	sHz	lAş	DHr	Hl	mA
şekillendirilebilir	that can be put into a certain format	şekil	lAn	dHr	Hl	yAbil	Hr	
akışkanlaştırıcılık	Having the property of making so mething fluid	Ak	Hş	GAn	lAş	DHr	HCH	lHk

Using synsets from WordNet Turkish (Ehsani et al., 2018) as source words, our linguists segmented words into meta forms, such as "_kitap_sev+Ar" (_book)(_love) for the word *kitapsever* (book lover). See Table 5.1 for examples

of words with multiple suffixes. Table 5.3 lists common suffixes in their meta form, while Table 5.2 provides frequency statistics on the distribution of suffixes in Turkish derived words.

Table 5.2 Number of number of suffixes.

# of suffixes	# of # of suffixes
6	2
5	28
4	327
3	2,169
2	9,373
1	16,618
0	19,954

Table 5.3 Most common suffixes.

Suffix	# of suffix
mAk	5,051
lHk	4,847
CH	3,384
lH	3,158
mA	2,266
sHz	2,200
lA	1,944
sH	1,836
lAş	1,535
CA	958
DHr	903
lAn	884
yHm	872
yHk	714
Hl	526
lAr	500
Hn	499
Ht	499
HcH	455
Hş	452

5.2 ANLAMVER DATASET

Unsupervised semantic modeling has recently attracted considerable interest within the NLP field, largely due to the adaptability of pre-trained models across various high-level NLP tasks such as word sense disambiguation, machine translation, and named entity recognition. The growing computational capabilities of unsupervised distributional semantic modeling (DSM) techniques allow researchers to improve NLP task performance by extracting semantic information from large amounts of unstructured text at relatively low costs. However, there is still a scarcity of methods and resources for intrinsically evaluating semantic models independently of the dynamics of high-level tasks. With the introduction of the AnlamVer dataset, which measures word similarity and word relatedness (i.e., association), we aim to provide the Turkish semantic modeling community with an intrinsic evaluation tool that tackles morphology-driven challenges brought about by the language's rich agglutinative structure.

While working on the AnlamVer dataset, we were unaware of the existence of any Turkish word relatedness or word similarity datasets in the literature. However, it turns out that the Sopaoglu (Sopaolu and Ercan, 2016) dataset, a relatedness dataset consisting of 101 word-pairs, with 65 of them translated from the RG dataset (Rubenstein and Goodenough, 1965), already existed in the literature. In this section, we outline the design principles and data collection guidelines adhered to during the creation of the dataset, along with a summary of the dataset's statistics.

5.2.1 Design Motivations

Word similarity evaluation, often referred to as wordsim, is among the earliest intrinsic methods for assessing semantic models. For instance, the RG dataset (Rubenstein and Goodenough, 1965) remains a widely used benchmark in DSM research to this day. Wordsim datasets are typically created by having human participants assign numerical scores, ranging from 0 to 10, to a set of predefined word pairs. In this section, we discuss the challenges associated with

word similarity evaluation and the design choices we implemented to address these issues in our study.

5.2.1.1 Similarity and Relatedness Confusion

Linguistic Background For over a century, linguists have been examining the statistical distributions of linguistic elements (words). While the distributional hypothesis, "words that occur in similar contexts tend to have similar meanings," is often attributed to Harris (1954), Sahlgren (2006) notes that the theoretical roots of this approach extend back to structuralist linguists Bloomfield (1887–1949) and Ferdinand de Saussure (1857–1913). De Saussure et al. (2011) emphasized the distinct functional roles that *signs* can have within a language system. He categorized the functional differences between linguistic elements into two (orthogonal) types, which are extensively studied in distributional semantics (DS) today: *syntagmatic* and *paradigmatic* relations. In brief, "words have a syntagmatic relation if they co-occur, and a paradigmatic relation if they share the same neighbors" (Sahlgren, 2006). Paradigmatic words represent similar concepts or entities in the real world and are often interchangeable within a given context. For instance, in the sentence "She is very [clever | smart]," the words *clever* and *smart* serve as synonyms and are unlikely to appear together in the same sentence.

Lack of Distinction in Word Evaluation: Although the theoretical difference between paradigmatic and syntagmatic relations can easily be applied to word evaluation by assuming that "similarity represents paradigmatic relations and relatedness represents syntagmatic relations," semantic research has not given this distinction the attention it deserves. Two of the most comprehensive DSM benchmark studies, (Baroni et al., 2014) and (Levy et al., 2015), assessed model performance using wordsim datasets like RG (Rubenstein and Goodenough, 1965), WordSim-353 (Finkelstein et al., 2001), and MEN (Bruni et al., 2012). In their work, Hill et al. (2016) thoroughly explain the limitations of such datasets due to the lack of clear distinction between similarity and relatedness (i.e., sim-rel). They also establish three criteria for evaluation

datasets: representativeness, **clear definitions**, consistency, and reliability. Most wordsim datasets, such as RG, MC (Miller and Charles, 1991), WordSim-353, and MEN, do not meet the clear definition criterion, as their screening guidelines use terms like "similarity," "relatedness," and "association" interchangeably. A prime example of this ambiguity is found in the instructions from the WordSim-353 study: *"...please assign a numerical similarity score between 0 and 10 (0 = words are completely unrelated), 10 = words are VERY closely related..."*. Since our study aims to collect both similarity and relatedness scores from participants, we provided clear and precise instructions in the questionnaire screens (Figure 5.1).

One Model Does Not Fit All Agirre et al. (2009) identified the sim-rel confusion and addressed it by dividing the original WordSim-353 dataset into two distinct sets: WS-Rel and WS-Sim, classifying word pairs based on their relationship types. This resolved the issue of sim-rel distinction in the dataset without needing to re-score the word pairs. They introduced two separate models for evaluating similarity and relatedness. For instance, they found that the context-window approach performs better at capturing similarity (evaluated on WS-Sim), whereas the bag-of-words approach excels at capturing relatedness (evaluated on WS-Rel). Capturing similarity seems notably more challenging for unsupervised models based on the *distributional hypothesis* than for relatedness models. In the DSM benchmark study by Levy et al. (2015), it is consistently shown that all model configurations perform the worst on the SimLex-999 similarity dataset (with an average of ≈ 0.39), compared to relatedness datasets (traditional wordsim datasets) (≈ 0.70).[22] [23]

Similarly, Hill et al. (2016) focuses solely on similarity evaluation, clearly distinguishing between similarity and relatedness for annotators in their SimLex-999 dataset. Another dataset, SimVerb-3500 (Gerz et al., 2016), concentrates on large-scale verb similarity evaluation, specifically targeting distributional verb

[22] Since participants provided scores under ambiguous guidelines, WS-Sim is not strictly a similarity dataset. See (Hill et al., 2016).
[23] Lower scores on the RW dataset are plausible due to its focus on out-of-vocabulary (OOV) and rare words.

semantics with 3,500 word pairs. We observe that DSMs are increasingly being tailored into more specialized models (e.g., relatedness, similarity, antonymy), driven by the need for improved performance in high-level tasks. As Faruqui et al. (2016) highlight, intrinsic wordsim evaluations often fail to align with the evaluation results of extrinsic NLP tasks. The confusion between similarity and relatedness in wordsim datasets may contribute to this discrepancy. It remains an open question whether a single pre-trained DSM can consistently represent domain-specific semantics across various high-level tasks. We believe that an ideal DSM would be a multi-model framework capable of handling different specific relationship types (e.g., relatedness, similarity, antonymy, hypernymy, meronymy) with optimal performance. In this study, our dataset targets Turkish-specific DSMs, focusing on two key semantic relations—similarity and relatedness—by evaluating word pairs across both dimensions simultaneously.

5.2.1.2 Out-Of-Vocabulary and Rare Words Problems

Turkish is an agglutinative language characterized by complex inflectional and derivational morphology. Its agglutinative structure allows the formation of new words by combining a stem with morphemes and suffixes. In Turkish, words consist of bound morphemes, meaning that each word can only have one lexical stem (root). Due to the large number of productive affixes (e.g., *CHk*, *CA*, *CI*, *lHK*, *SHz*, *HmsH*), an infinite number of surface forms can theoretically be generated. Table 5.4 illustrates the inflections and derivations of various words in their morphologically decomposed forms, all sharing the lexeme *maymun* (monkey). In this study, all morphological decompositions were conducted using the toolkit from Görgün and Yıldız (2011).

Table 5.4 Morphological decomposition of various words sharing the same lexeme.

Word	Decomposition	Sense	Form	Frequency
maymun	maymun	monkey	root form	very
maymunları	maymun + lAr + sH	their monkeys	inflectional	medium
maymunsu	maymun + sI	ape, like monkeys	derivational (usual deviance)	rare
maymungilleri	maymun + gil + lAr + yH	family of monkeys, primades	derivational (acceptable deviance)	oov
maymuncuk	maymun + CHk	skeleton key, picklock (tool)	derivational (deviant)	rare

Simple word-based models overlook the internal structure of words, which limits their overall capability and effectiveness. A major challenge arises when a word in the test set either has not been encountered during training (i.e., OOV) or has appeared only infrequently (i.e., rare-words). To address these issues, the distributional semantics (DS) community has been developing more advanced subword-level (i.e., compositional) models. These models are designed to better handle the OOV and rare word problems, especially in morphologically-rich languages, where DSMs need to manage such challenges to achieve better generalization. The RW dataset (Luong et al., 2013) introduces a word-frequency-based evaluation strategy to help developers of compositional models deal with word rareness. Similarly, our goal is to balance the word pool in our dataset based on word frequencies, in order to evaluate the generalization capabilities of these models effectively.

In addition to the conventional evaluation strategy for OOV and rare words, we introduced a new concept called *made-up words* by adding *novel* (i.e., invented or fictitious) words into our dataset's word pool. Vecchi et al. (2017) applied a similar idea to their phrase-level models to assess the creative abilities (i.e., generalization power) of the models. The core idea is that, even when people encounter a word for the first time, it may sound unfamiliar, but they still possess the intuition to infer its intended meaning. Can DSMs achieve this as well? In our case, working at the subword level, we hypothesize that Turkish affixes can consistently alter word meanings, a phenomenon referred to as *acceptable semantic deviance*. For instance, while the word *maymungilleri* (family of monkeys) is fabricated and may sound unusual to a native Turkish

speaker (Table 5.4), most speakers can grasp its meaning to some degree. This linguistic productivity highlights a significant potential for model generalization in research. However, the downside is that this assumption does not always hold, as seen with the word *maymuncuk* (skeleton key, a tool) (Table 5.4). In this case, although the word is derived from the root *maymun* (monkey) using the *cHk* affix in a valid morphological process, one of its senses shifts to an entirely different semantic domain. This type of semantically lossy derivation poses a significant challenge for compositional DSMs, which Vecchi et al. (2017) refer to as *deviants*.

5.2.1.3 Dataset Translation Issues

Before beginning the dataset construction phase, we initially considered translating well-known wordsim datasets into Turkish. However, after translating the MC dataset, we determined that creating a new dataset from scratch would be more meaningful and reliable than relying on translations of existing datasets. The primary reasons for this decision can be summarized as follows:

- Both words in the original word-pair correspond to the same single word in the target language: "*football - soccer*" , →, "*futbol - futbol*".
- One word in the source word-pair translates to a multi-word phrase: "*asylum - madhouse*" , →, "*tımarhane - akıl hastanesi*". Traditional wordsim datasets and DSMs typically exclude phrases for the sake of simplicity in modeling and evaluation, and we have similarly chosen to exclude phrases from the scope of this study.
- Translation-related shifts in meaning require re-annotation by human evaluators for each word-pair. The human annotation process is one of the most resource-intensive stages of the study.[24]
- Our goal is to balance words and word-pairs across as many attributes as possible, including word frequency, derivation, inflection, concreteness,

[24] This includes costs related to human resources, questionnaire software, and data pre/post-processing.

and relation types. Word frequencies and morphological features are heavily dependent on the specific language.

5.2.2 Dataset Construction Pipeline

The design motivations, as outlined in the previous section, can be summarized as follows: (i) gathering two-dimensional relatedness and similarity scores from participants while clearly defining the distinctions between these concepts, (ii) creating a language-specific morphological dataset to evaluate DSMs' generalization abilities with respect to OOV, rare words, and semantic deviance scenarios, and (iii) balancing the dataset as much as possible across multiple morphological and semantic attributes. Given the time and budget constraints, we set the target dataset size at 1,000 scores (500 word-pairs), which is in line with or larger than most existing wordsim datasets (e.g., SimLex-999=999, RG=65, MC=30, WordSim-353=353, RW=2,034, MEN=3,000). We organized the dataset construction process into three phases: word candidate selection, word pool creation, and word-pair selection (Table 5.5).

Table 5.5 Morphological decomposition of various words sharing the same lexeme.

	Stage 1	Stage 2	Stage 3
	1) Word Candidates (starts)	2) Word-Pool Selection	3) Word-Pairs Selection
Goals	1.1) Reusing existing resources	2.1) Balancing word attributes by estimations	3.1) Balancing word-pairs by estimations
Input	1.2) TKN (600) + MC (39)	2.2) Stage1 output (639) + new derivational words (99)	3.2) 320 Stage2 words
Process	1.3) Attaching frequencies, morphological tags	2.3) Filtering for balancing	3.3) Mapping pairs (every word used 2-5 times building word-pairs)
Output	1.4) 639 words	2.4) 320 words	3.4) 500 word-pairs (ends)

5.2.2.1 Word Candidates Selection

TKN Dataset Since Turkish is considered a low-resource language in NLP research, our goal was to maximize the re-use of existing resources. We examined word candidates that already included useful attributes, utilizing the

"Türkçe Kelime Normları" (TKN) dataset (Tekcan and Göz, 2005), originally developed for a psycholinguistic study. The TKN dataset contains 600 Turkish words, balanced by their *concreteness* values, with half of the words being concrete and the other half abstract. These concreteness values were annotated by 100 voluntary university students. Similar to the USF word norms dataset in English (Nelson et al., 2004), TKN's concreteness scores range from 1 to 7, with lower values indicating more abstract concepts and higher values indicating more concrete ones. For example, the word *mutluluk* (happiness) has a score of 1.85, while *gül* (rose) scores 6.79. By selecting candidates from the TKN dataset, we ensured that model developers could evaluate their models across different concreteness levels. Unfortunately, TKN primarily contains frequent, root-form words, with 480 in root form and none of the remaining 120 being inflected.[25] To address these limitations and maintain dataset balance, we manually added 99 additional words (without concreteness values) in the next stage.

5.2.2.2 Word-pool Selections

With 600 candidate words carried over from the initial stage, our objective was to reduce this number to 320 words (word-pair candidates) while ensuring that the dataset remained balanced according to our criteria. Table 5.6 presents the grouping attributes for the word pool, along with the corresponding word counts and percentages.

Frequency-based Balancing Given the importance of addressing OOV and rare word challenges in models for morphologically rich and productive languages, our primary focus was to balance our word pool based on word frequencies. The RareWords dataset (Luong et al., 2013) tackles this issue by classifying words into four frequency ranges ($5-10$, $10-100$, $100-1000$, $1000-10000$). However, since the RareWords dataset is tailored for English, a language that is less inflectional and productive compared to Turkish, researchers might assume that words with frequencies below five are likely non-

[25] 108 words have one derivation, while 12 have two derivations.

English or junk words. In contrast, a single Turkish lexeme can generate thousands of surface forms. Our corpus coverage analysis shows that 47% of word types (277K) appear only once in the corpus, which aligns with the word coverage statistics from the Boun Corpus (Sak et al., 2011). Therefore, it was essential not to overlook words with zero or fewer than five occurrences. We employed a different grouping strategy, with the first group consisting of OOV (zero frequency) words, and the remaining words divided into five rare word groups (RW 1, RW 2, RW 3, RW 4, RW 5). Table 5.6 shows how OOV and rare words are distributed across the word pool. We conducted the frequency analysis using the Boun Corpus (Sak et al., 2011), which contains approximately 3.2 million token types (i.e., vocabulary size). We defined the frequency ranges (0 − 32, 32 − 320, 320 − 3200, 3200 − 32000, 32000 − ∞) using the function gr(n, voc, g), where g is the number of groups, n is the group index (ranging from 1 to g), and voc is the corpus vocabulary size. The minimum and maximum values for the first and last groups were fixed at 0 and ∞, respectively. Ampersand symbols (&) denote string concatenations:

$$gr(n, voc, g) = (voc \times 10^{-(g-n+3)}) \text{ \& "-" \& } (voc \times 10^{-(g-n+2)})$$

Table 5.6 Groupings of the word-pool.

	G0	G1	G2	G3	G4	G5	Total
Frequency	OOV	RW1	RW2	RW3	RW4	RW5	
	31 9.6%	33 10.3%	30 9.3%	62 19.3%	111 34.6%	53 16.5%	320 100%
Concreteness	no value	abstract	medium	concrete			
	149 46.5%	35 10.9%	30 9.3%	106 33.1%			320 100%
Root Form	root	non-root					
	182 56.8%	138 43.1%					320 100%
Derivations	no der.	der1	der2+				
	198 61%	81 25.3%	41 12.8%				320 100%
Inflections	no inf.	inf1	inf2+				
	277 86.5%	17 5.3%	26 8.1%				320 100%

5.2.2.3 Word-pair Selections

During the word-pair selection phase, we combined words from the word pool to create new pairs, with the constraint that each word could appear in up to five different word-pair combinations. The main objective of this stage was to generate 500 word-pair relationships that were explicitly balanced across new attribute types, such as estimated semantic relations (e.g., synonym, antonym, hypernym, meronym). We manually matched the words and estimated the semantic type of their relationships. For instance, we selected the words *otomobil* (automobile) and *araba* (car) from the pool, marking them as a strong synonym match. We categorized 50 pairs as synonyms, 50 as antonyms, 50 as meronyms, and 50 as hypernyms. Additionally, we grouped the word pairs by their estimated degrees of relatedness (low, medium, high). Table 5.7 presents the actual instances and percentages for these estimation-based and morphological groupings of word pairs. In total, we finalized 500 manually

selected and grouped word pairs. Table 5.8 provides examples of annotated word-pair instances from the completed dataset.

Table 5.7 Groupings of the word-pairs.

	G0	G1	G2	G3	G4	G5	Total
Est. Synonyms	synonym	antonym	other				
	50 10%	50 10%	400 80%				500 100%
Est. Relatedness	high	medium	low				
	200 40%	150 30%	150 30%				500 100%
Est. Rel. Type	hyponym	meronym	other				
	50 10%	50 10%	400 80%				500 100%
OOV	no oov	any oov	two oov				
	434 86.8%	66 13.2%	42 8.4%				500 100%
Min. Derivations	no der.	der1	der2+				
	231 46.2%	166 33.2%	103 20.6%				500 100%
Min. Inflections	no inf	inf1	inf2+				
	424 84.8%	32 6.4%	44 8.8%				500 100%
Min. RareWord	rw0 (oov)	rw1	rw2	rw3	rw4	rw5	
	66 13.2%	65 13%	62 12.4%	130 26%	142 28.4%	35 7%	500 100%

5.2.2.4 Questionnaire Design

Platform We developed a web-based application to gather data from human annotators. Participants were asked to rate the similarity and relatedness of 500 word pairs, providing 1,000 responses in total (two scores per word pair). The questionnaire was divided into two sections. The first section introduced the concept of similarity with a detailed explanation, accompanied by five examples. Following the SimLex-999 guidelines, participants were instructed to assign low scores for antonyms and high scores for synonyms. We defined similarity as

follows (excerpt from the first two sentences): *"Two words are considered similar if they refer to the same object, person, concept, or action. Similar items share common concrete or abstract attributes. For example, 'tea' and 'coffee' are quite similar because both are comforting hot beverages derived from nature and are irreplaceable companions in social conversations."* Figure 5.1 shows a snapshot of the initial guideline screen for similarity annotation. Upon clicking the "ileri" (next) button, participants were presented with the first word-pair page, where they rated 20-word pairs per screen.

Table 5.8 Sample word-pairs from the final dataset. Words with asterisks (*) are made-up words. (adh = adherent, conc = concreteness, ss = semantic sub-space, syn = synonyms, ant = antonyms, irr = irrelevant, der# = total derivations, inf# = total inflections)

word1	word2	sim.	rel.	oov	conc.	ss	der#	inf#
otomobil (automobile)	araba (car)	9.16	9.33	no	6.87	SR (syn.)	0	0
üşengen* (lazy)	üşengeç (lazy)	8.25	7.83	one	3.06	SR	2	0
atatürkist* (adh .to Atatürk)	kemalci (adh. to Kemal)	8.75	9.63	one	-	SR	2	0
kitaplıklar (bookshelves)	kitaphane* (place with books)	7.16	8.41	no	-	SR	2	1
kemalci (adh. to Kemal)	kemalizmcilerden (from ...)	5.25	8.66	one	-	SR	3	3
kırmızı (red)	gül (rose)	1.16	7.16	no	6.79	DR	0	0
şeffaf (transparent)	opak (opaque)	1.16	7.16	no	4.37	DR	0	0
zarar (loss)	kazanç (profit)	0.18	8.8	no	3.25	DR (ant.)	0	0
gevşek (loose)	heykel (statue)	0.16	0.16	no	-	DU (irr.)	0	0
üşengen* (lazy)	yedigen (heptagon)	0.16	0.25	two	-	DU	2	0

Participants All 12 participants were native Turkish speakers who voluntarily took part in the questionnaire. The group included eight females and four males, with both the mean and median ages being 33.5 years, and a standard deviation of 9.3. Nine participants held university degrees (seven with a master's degree), two were undergraduates, and one participant had a high school diploma. Participants were invited to take part remotely by following an invitation link sent to their email. Thanks to the responsive layout of the WSQuest questionnaire software[26], they could easily complete the task on mobile or tablet devices. Participants were instructed to carefully read the user

[26] For full user screen guidelines, refer to the appendices or visit http://www.gokhanercan.com/wsquest

guidelines, as none had prior experience with the annotation process or the concepts of word similarity and relatedness. The initial guideline screen informed users that they could complete the questionnaire at any time over a three-day period, with the option to resume their session as long as they saved the last completed URL. On average, it took 75 minutes for participants to finish the entire questionnaire without taking breaks.

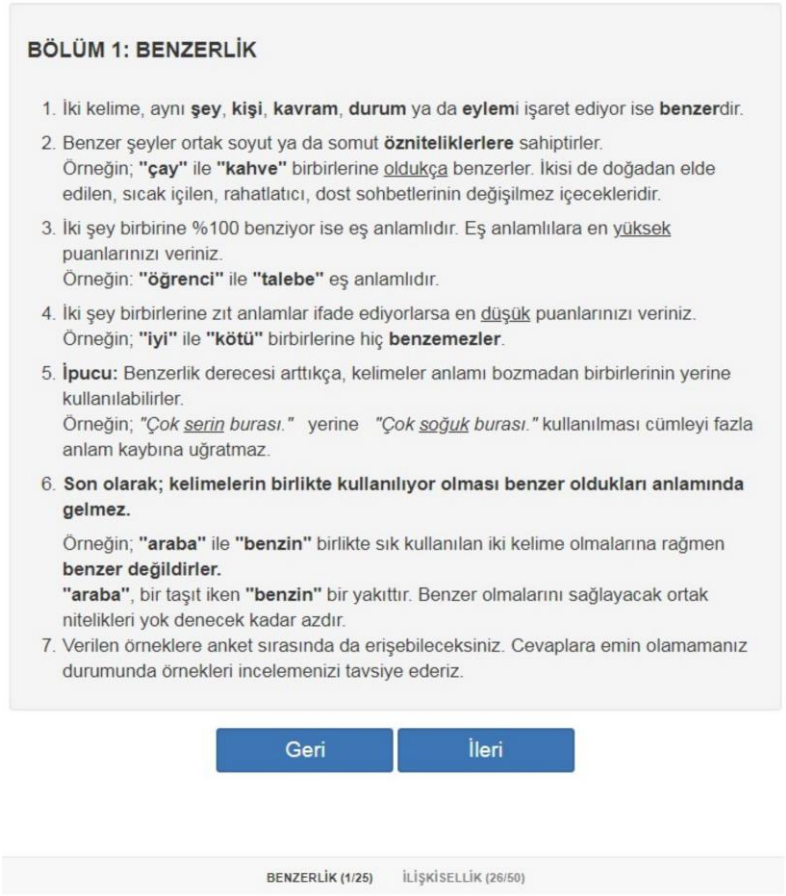

Figure 5.1 Similarity instructions page.

Figure 5.2 Word-pair annotation page.

5.2.3 Dataset Analysis

The final (actual) similarity s and relatedness r values in the dataset appear to align with our initial estimates. Based on the assumptions and configuration of the Sim-Rel vector space model, the scatter plot of the average s and r values produced a visual outcome that closely matched our expectations (Figure 5.3). Our key observations regarding the actual data distribution are as follows:

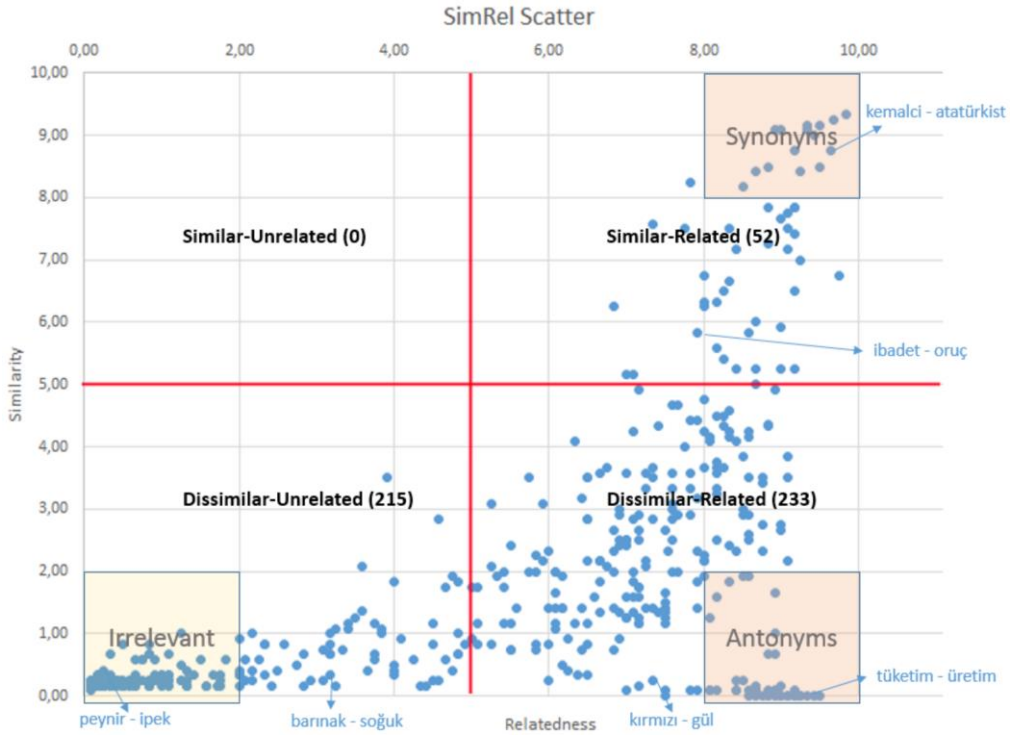

Figure 5.3 Scatter plot of the final AnlamVer dataset. Data-points denote participants' avg. Sim-Rel score of each word-pair where y axis is s and x axis is r. Member counts of ss{SU, SR, SU, DR} semantic sub-spaces are in the parenthesis.

- As expected, the SU subspace remained empty. Participants confirmed that word pairs cannot be both similar and unrelated simultaneously.
- The antonym-DR overlap issue persisted, with antonym and DR word pairs receiving very similar scores. For instance, the word pair *kırmızı - gül* (red - rose) had average s and r scores of 1.16 and 7.16, respectively. Similarly, the antonym-estimated pair *şeffaf - opak* (transparent opaque) received identical scores to the former (see Table 5.8).
- Participants treated word pairs containing made-up words as they would regular word pairs. For example, the pair *atatürkist - kemalci* (*atatürk+İST - kemal+CH*) was rated as 8.75 for similarity and 9.63 for relatedness (see Table 5.8), even though neither surface form is commonly used in Turkish (OOV and RW1, respectively). The suffixes *İST* and *CH* are used to denote "ideological adherence to a person/thing,"

with *Atatürk* and *Kemal* referring to Mustafa Kemal Atatürk, the founder of the Republic of Turkey.

5.2.3.1 Post-processing and Inter-annotator Agreement

Since the questionnaire included many uncommon OOV and rare word pairs, participants were given the option to leave a word pair unanswered if they were unsure of its meaning. However, the null response rate was much lower than anticipated, at just 0.1%. To ensure accurate calculation of ranking correlations, we replaced any null responses with the average score provided by all other participants for that question.

Out of the 16 participants, we conducted post-analysis on the collected data and found that one participant had a notably low Spearman ranking correlation score (0.32 min, 0.57 max) compared to the others. Upon further review, we discovered that this participant completed the questionnaire in just 25 minutes, which is three times faster than the estimated time for high-quality annotations. Similarly, we excluded the responses of three additional participants to enhance the overall data quality.

After post-processing, the average pairwise inter-annotator agreement (*apia*) score was **0.748**, with the highest pairwise correlation being 0.847 and the lowest at 0.474. Although the dataset's *apia* score is slightly lower than expected, 0.748 is still higher than the inter-annotator scores for many other word similarity datasets (WS-Sim=0.667, MEN=0.68, SimLex-999=0.67). According to Snow et al. (2008), using more than ten annotators is considered statistically acceptable for ensuring the reliability of a word similarity evaluation task.

5.2.4 Summary

We introduced a semantic model evaluation dataset specifically for the Turkish language, which, due to its complex morphology, necessitates advanced semantic models to address OOV and rare word challenges. With 13% of the dataset consisting of OOV pairs and 26% of rare word pairs (RW1 and RW2),

we believe it will provide a demanding intrinsic evaluation task for DSM researchers. Our hope is that models evaluated using AnlamVer, with its distinct similarity and relatedness measures, will show improved correlations with higher-level NLP tasks.

5.3 OSIMUNR DATASET

5.3.1 Design Motivations

Irrespective of whether they measure similarity or relatedness, conventional wordsim datasets (i.e. the wordsim task) such as WordSim353 (Finkelstein et al., 2001), RG (Rubenstein and Goodenough, 1965), MG (Miller and Charles, 1991) have long served as one of the main performance measures for DSMs alongside the analogy task (Mikolov et al., 2013d). Both tasks are widely adopted due to their high reusability (i.e., task-independent) and relatively straightforward construction. While they are mostly considered as intrinsic evaluation methods for DSMs (Gladkova and Drozd, 2016; Faruqui et al., 2016; Hadj Taieb et al., 2020), it is important to note that they rely on external human annotations collected as answers to specific set of questions. We argue that the word similarity/relatedness datasets and the wordsim task itself lack testing the problems *noise*, *overlapping n-grams*, and *orthographic similarity* correlation which are the primary focus of this study. As suggested by Gladkova and Drozd (2016): ”a shift from abstract ratings of word embeddings quality to exploration of their strengths and weaknesses”, below, we outline some problems about how such methods and datasets fail to identify the aforementioned weaknesses of DSMs.

5.3.1.1 Tasks Measure Relative Relationships, not Absolute Values

The wordsim task simply measures the ρ Spearman (1961) rank correlation between the model predictions and the human annotation scores of all word-pairs within the dataset. The Spearman ranking correlation is ideal for measuring the

relative semantic performance of DSMs since it measures the ranking correlation of scores instead of measuring the actual *absolute* values. This relativity perfectly handles inconsistencies between annotators by softening annotators' subjective scoring scales. For example, it corrects one annotator's unusual behavior, such as not scoring lower than 2/10 even for the most unrelated word-pairs, such as *cord – smile*. Since most people think that the *cord – smile* word-pair should have a score very close to 0 on average (its final average score is 0.02/4 in the RG dataset), the Spearman correlation can help mitigate scaling inconsistencies as long as the rankings are similar. Thus, it is also ideal for calculating the inter-annotator agreement score of datasets. However, when correlating human scores with model predictions, if the model space is somehow *skewed* (Figure 6.1 and 6.2), it could conceal the abnormal value predictions made by the models. For instance, suppose a model predicts moderately high relatedness scores as 6/10 for almost all unrelated word-pairs (actual FT score for *cord – smile* is 0.57), the wordsim task cannot detect this abnormality when the rankings of word-pairs are relatively correlated well with the rankings of human scores.

Our experiments confirm this scenario, where Char-gram[3-6] segmentation consistently yields higher scores for every word-pair than expected while getting similar results from wordsim task (ρ_{rel} = 0.61, ρ_{RG} = 0.77, Table 6.7). For an NLP application that requires absolute relatedness scores for given word-pairs (e.g., semantic word usage checker tool), it would be unacceptable to get a score of 5.7/10 for the *cord – smile* word-pair. It should be noted that, in OSimUnr cases, scores can be high as 8.1/10 for totally unrelated concepts such as *adventure – denture*, which cannot be identified by **relative** evaluation methods (see FT column in Table 2.6).

5.3.1.2 Distributional Mismatch

Since there is no consistent methodology for selecting word-pools and word-pairs for the construction stage of wordsim datasets (Hadj Taieb et al., 2020), datasets tend to vary in relation types, POS constraints, word frequencies,

morphological forms of words, and other factors. Moreover, dataset sizes are often quite limited to cover special cases like OSimUnr. According to the survey study by Hadj Taieb et al. (2020), among the 51 datasets it covers, MEN (Bruni et al., 2014) is still the largest word relatedness dataset for English, containing 3,000 word-pairs. Most of the similarity/relatedness datasets are smaller than 1,000 word-pairs, with an average of 405 word-pairs across 19 relatedness datasets. Consequently, many wordsim datasets primarily cover common word-pair scenarios, potentially overlooking special cases in evaluation.

Table 5.9 Average orthographic similarities and lengths of some existing wordsim datasets. Original wordsim dataset scores and orthographic similarity scores are normalized to scale 0-10. * Since AnlamVerOOV is a subset of AnlamVer dataset, it is excluded from the mean calculations. Scale: original scale of the dataset, Len: average string length/word, Score: average rel/sim score of word-pairs. See §3.4.1 for editsim(ES) and Ngram[3](NG) measures.

Dataset	Type	Scale	Size	Len	Score	ES	NG
MC (Miller and Charles, 1991)	rel	0-4	30	5.50	4.92	1.21	0.96
RG (Rubenstein and Goodenough, 1965)	rel	0-4	65	5.80	4.69	1.13	0.89
WS353 (Finkelstein et al., 2001)	rel	0-10	353	6.49	5.86	1.42	1.25
RareWords (Luong et al., 2013)	rel	0-10	2,034	8.75	6.21	2.24	1.94
MEN (Bruni et al., 2014)	rel	0-50	3,000	5.50	5.00	1.32	1.17
MTurk771 (Halawi et al., 2012)	rel	1-5	771	6.14	7.4	1.29	1.09
SimLex-999 (Hill et al., 2016)	sim	0-10	999	5.64	4.56	1.44	1.35
English Mean	-	-	1,036	6.27	5.51	1.44	1.23
AnlamVer (Ercan and Yıldız, 2018)	both	0-10	500	6.78	4.98	1.53	1.47
AnlamVerOOV* (Ercan and Yıldız, 2018)	both	0-10	66	10.98	4.73	2.77	2.50
Sopaoglu (Sopaoglu and Ercan, 2016)	rel	0-5	101	5.60	5.75	0.98	1.06
WordSimTr (Üstün et al., 2018)	sim	1-10	140	10.68	4.95	5.62	4.95
Turkish Mean	-	-	247	7.69	5.23	2.71	2.50
Overall Mean	all	-	641.5	6.98	5.27	2.07	1.86

The structural mismatch between the existing relatedness datasets and the OSimUnr problems can be attributed to three main factors: Firstly, the limited sizes of most datasets (e.g., MC, RG, WS353) result in a bias towards including only very frequent word-pairs (e.g., *car – automobile*). Secondly, as reported by the study from Zesch and Gurevych (2006), authors tend to choose words that are related (e.g., *brother – lad*) during the word-pairing stages, shown in Table

5.9 and Figure 3.4 (TR-AVG=5.23/10, EN-AVG=5.51/10). This distributional bias significantly reduces the likelihood of word-pairs conforming to the OSimUnr case.

The third mismatch with the existing datasets is that the word-pairs have relatively short string lengths and are not particularly orthographically-similar. As shown in Table 5.9 and Figure 3.4, the average word length is 6.98 (tr=7.69, en=6.27) for widely used wordsim datasets covered in this study. Since the average orthographic similarity scores of word-pairs are approximately 2.5 for Turkish and 1.5 for English datasets on a 0-10 scale (see editsim and Ngram[3] columns in Table 5.9), wordsim datasets are far from covering orthographically-similar word-pairs scenarios. This distribution of word-pairs might seem natural, but it falls short in testing DSMs against the weakness of *orthographic sensitivity*. The average word lengths of Turkish datasets (and the RareWords dataset for English) are slightly greater than the others because researchers intentionally chose word-pairs in derivational and inflectional forms to challenge models against OOV and rare-word problems. As a result, the orthographic similarity scores tend to increase due to the co-occurrence of common derivational and/or inflectional affixes in word-pairs, as exemplified by the word-pair '*_konuş+kan+**lığ+ı+na*** – *_çene+baz+**lığ+ı+na***' by the AnlamVer dataset. Considering the average number of morphemes per word (i.e., index of synthesis) for the Turkish language is almost two times higher (tr=2.86, en=1.68) than for the English language (Karlsson, 1998), the *orthographic sensitivity* problem becomes more significant for more synthetic languages such as Turkish, Russian, or Finnish. Even though English does not have rich inflectional morphology as Turkish, its derivational nature is also prone to generating orthographically-similar but unrelated words. By employing *fully derivational* segmentation methods (e.g., *_act+ive+ate+ion*), we managed to achieve thousands of orthographically-similar but unrelated word-pair scenarios such as '*_canon+**ize+ion*** – *_carbon+**ize+ion***' for the English language as well.

5.3.2 Construction Pipeline

We designed a dataset construction pipeline for automatically building OSimUnr word-pairs in four main stages (Table 5.10). The same processing pipeline is applied to both Turkish and English languages. We publicly release the dataset construction outputs of each stage as separate data files.[27] Our dataset construction pipeline does not contain human intervention, except for the sub-stage Categorical Filters (see §5.3.2.4). In this sub-stage, we apply type, type-pair, and affix blacklist exclusions defined by the researchers. This step is included to provide an additional layer of error reduction in the final dataset. Since all outputs of the subsequent stages are constructed automatically based on predefined constraints, the final dataset is free from human biases in word selection, word-pairing and relatedness scoring. Consequently, the pipeline process is deterministic and reproducible, as it does not introduce randomness at any selection points. We acknowledge that the automatic nature of our pipeline exhibits **resource bias**, which encompasses all the tools and datasets in our tool stack along with their inherent bugs, biases, and the limitations of our implementation capabilities.

Calculating an error rate for the OSimUnr dataset is not a straightforward task. Considering the sheer volume of nearly a million word-pairs, the subjective task of labeling word-pairs as related or unrelated is not practical for humans to address without referring to external sources. Despite our efforts to minimize errors in the dataset through the described steps, we are unable to scientifically report an error rate based on human ground truth.

[27] https://github.com/gokhanercan/OSimUnr or http://gokhanercan.com/OSimUnr

Table 5.10 Four main stages of the dataset construction pipeline.

#	Stage	Input Type	Output T.	Output Sample
1	Word-pool Selection	[Word sources]	Words	..., crammer, .., gramma, grammar, ...
2	Word-pair Matching	Words	Word-pairs	grammar – crammer (osim=editsim=0.71)
3	WN Relatedness Appr.	Word-pairs	Word-pairs	grammar – crammer (rel=lch=0.22)
4	Relatedness Filtering	Word-pairs	Filter in/out	Add to OSimUnr Q3 editsim dataset

5.3.2.1 Word-pool Selection

The first stage is word-pool selection, which aims to automatically select word candidates from existing resources based on certain word filtering constraints, rather than manually hand-picking them. As initial word sources for the pipeline, we employ WordNet 3.0 (Miller, 1995) through the Python implementation NLTK (Bird et al., 2009) for English. For Turkish, we utilize WordNet KeNet (Bakay et al., 2021) along with its Java implementation.[28] We include only single words by filtering out phrases (e.g., *political theory*) and words with hyphens (e.g., *ill-smelling*). We exclusively **incorporate nouns** (i.e., N) (including proper nouns) into the dataset, primarily to enhance simplicity and facilitate WordNet hierarchies. WordNets demonstrate exceptional proficiency in representing taxonomic IS-A relations, such as hypernymy and hyponymy, specifically for nouns (e.g., *car* → *vehicle* → *entity*) in noun-to-noun (i.e., N-N) matchings. Conversely, adjectives (i.e., A) lack a comparable organization in IS-A relations (Pedersen et al., 2004); hence, we deliberately excluded them to mitigate potential errors.

Despite WordNets' support for verb (i.e., V) relationships, the morphological analysis and disambiguation of verb derivations present significant challenges for Turkish. For instance, the most atomic roots that derive verbs are very short (e.g., *kur, bas, tut, at, ol, el, sür*), and they are derived with short derivational suffixes, primarily consisting of commonly used vowels (e.g., *+A, +A(C), +A(l), +A(K), +I*). Moreover, a significant portion of these verb derivations has lost their productivity throughout the evolution of language,

[28] https://github.com/olcaytaner/TurkishWordNet v1.0.49

limiting their applicability to only a limited number of roots. The presence of such short meta affixes results in a multitude of morphological parse candidates, subsequently increasing the likelihood of errors during the disambiguation process. More importantly, **WordNet does not cross part-of-speech boundaries** (Pedersen et al., 2004) when establishing relationships, which renders the modeling of even seemingly trivial relatedness relations between *drink* (V), *red* (A), and *wine* (N) challenging. As a result, we decided to exclude verbs and adjectives from the dataset. These exclusions aim to ensure dataset quality and simplify the morphological analysis process.

Table 5.11 Data flow through dataset construction pipeline. Numbers indicate the final number of items (words for stage 1, word-pairs for stages 2 and 4) yielded from each stage. Q3+Q4 denotes combined dataset where orthographic similarity scores are between 0.5 and 1.

St.	Dataset Construction Stages	English		Turkish	
1	**Word-pool Selection**				
	Word-pool Source (reference)	English WordNet 3.0 (Miller, 1995)		Turkish WordNet KeNet (Bakay et al., 2021)	
	WordNet Implementation	NLTK (Bird et al., 2009)		Java Lib.[7]	
	Initial WordNet Size (Lemmas)	147,306		80,942	
	POS Filtered (Nouns only)	57,506		48,560	
	MinLenght >=6 and Punc. Filtering	46,634		24,952	
2	**Word Pairing**	**editsim**	**over_ft23**	**editsim**	**over_ft23**
	Possible Word-pairs ($(n^2)/2$ matchings)	$46{,}634^2/2$		$24,953^2/2$	
	Orthogr. Similar Word-pairs (Q3 + Q4)	4,674,094	1,117,717	2,057,834	424.450
	a) $Q3$ $(5 <= OSim(w_1, w_2) < 7.5)$	4,619,520	1,080,368	2,026,929	406,497
	b) $Q4$ $(7.5 <= OSim(w_1, w_2))$	54,574	37,349	30,905	17,953
3	**WordNet Relatedness Approximation**	`lch`		`wup`	
4	**Relatedness Filtering**	**editsim**	**over_ft23**	**editsim**	**over_ft23**
	Orthogr. Similars (OSimBinary-Q4)	53,771	-	30,905	-
	Orthogr. Similar But Unrelated (Q3 + Q4)	570,172	69,821	333,963	38,596
	a) $Q3$ $(5 <= OSim(w_1, w_2) < 7.5)$	567,457	68,672	332,119	38,057
	b) $Q4$ $(7.5 <= OSim(w_1, w_2))$	2,715	1,149	1,844	539

Another constraint we applied to word-pools is the minimum word length. As our analysis (see §3.2.1) on existing datasets suggests *lengthy words tend to be more sensitive to orthographic similarity*. Therefore, we included only the more error-prone lengthy words, by setting the minimum length to six. This setting also enabled us to minimize the size of the word pools before the word-

pair matching stage, which exhibits quadratic complexity in word-to-word matchings. After applying all filters, the final word-pools were reduced to 24,952 from 80,275 words for Turkish, and 46,634 from 147,306 words for English (see Table 5.11).

5.3.2.2 Word Pairing

In the second stage, we exhaustively take every word from the word-pools and test their matchings with other words to build up the word-pairs that fit our predefined orthographic similarity condition by editsim or over_ft23 measures. We only accept word-pairs if their orthographic similarity scores are greater than 0.5/1 (Equation 3.3). Since the complexity of the matching process is quadratic ($O((n/2)^2)$), it would normally take about a week to execute matchings per language on a standard computer.[29] We once again cythonized our Python implementation to reduce computation time. The final execution took approximately 12-16 hours per language. We organize the final orthographically-similar word-pairs into two groups based on their scores. We denote orthographically-similar word-pairs as Q4 when the orthographic similarity score is greater than or equal to 7.5/10. Word-pairs with *moderate* scores between 5/10 and 7.5/10 (5 <= OSim(w1, w2) <= 7.5) are denoted as Q3. Finally, for the editsim sub-dataset, the word-pair matching stage resulted in 54,574 word-pairs in group Q4 for English and 30,905 word-pairs for Turkish (Table 5.11). In group Q3, as expected, the process yielded millions of word-pairs that are moderately similar, such as the word-pair *unprocurable – unproductive* with an orthographic similarity score of 5.8/10. It should be noted that some of the generated orthographically-similar word-pairs represent identical concepts (e.g., *verbalizer – verbaliser*) or related concepts (e.g., *academia – academic*), while others are entirely unrelated (e.g., *action – auction*, *poison – prison*). Since we only need unrelated instances, we will eliminate the related word-pairs by leveraging WordNet relatedness approximations and

[29] Python 3.6 on Microsoft Windows 7, 16 GB Memory, Intel Core i7 2.60 GHz, SSD.

derivational morphology at the fourth stage of the pipeline, Relatedness Filtering (§5.3.2.4).

5.3.2.3 WordNet Relatedness Approximation

The previous stage yields millions of word-pairs (≈ 5.7M for English, ≈ 2.4M for Turkish), which are expected to be filtered and categorized by relatedness detection methods in subsequent stages. Instead of obtaining relatedness judgments from humans for millions of records, which can be a resource-intensive operation, we leverage existing WordNet relatedness/similarity methods to approximate relatedness. This enables us to use approximated scores for the tasks we propose: *unrelatedness-identification* and *relatedness-classification*, which involve labeling given word-pairs as *related*, *unrelated* or both. Unlike the conventional wordsim evaluation that requires highly precise relatedness/similarity scores, our approach does not depend on such exact values. While the WordNet-based approximation methods may not yield scores accurate enough for strong ranking correlations, we presume that they possess sufficient sensitivity to correctly label a word-pair as related or unrelated. Our primary objective in this phase is to identify the most suitable approximation methods for each language, which can simulate human relatedness judgments with the least error. To measure these approximation errors, the common practice is to use existing wordsim dataset scores as the ground-truth.

Approximation Methods We employ six (three for Turkish) WordNet-based methods at our disposal: wup (Wu and Palmer, 1994), path (Pedersen et al., 2004), lch (Leacock and Chodorow, 1998), lin (Lin et al., 1998), jcn (Jiang and Conrath, 1997), res (Resnik, 1995). These methods are often referred to as similarity measures (Pedersen et al., 2004) rather than relatedness (Budanitsky and Hirst, 2006; Zhang et al., 2013). These methods define path distance based formulations to approximate similarity/relatedness by incorporating IS-A relationship nodes (synsets) of WordNet databases (Equation 5.1,5.2). For example, wup similarity is a normalized measure calculated by dividing the

global depth of the *longest common ancestor* of concepts (i.e., lcs) by the total depth of two concepts (c1 and c2 in equations).

Figure 5.4 WordNet IS-A type graph depicts how related concepts can be distant in path distance.

In an attempt to enhance performance, lin, jcn, and res methods employ an information-based approach by combining path-based calculations with corpus-driven count-based TF/IDF models (our NLTK implementation uses Brown corpus), known as information-content (i.e., IC).

$$wup(c1, c2) = 2 \times \frac{depth(lcs(c1, c2))}{depth(c1) + depth(c2)} \tag{5.1}$$

$$lch(c1, c2) = -log \frac{len(c1, c2)}{maxdepth(c), c \in WordNet} \tag{5.2}$$

Instead of *words*, WordNets represent concept relationships through *synsets* (i.e., senses), which can encompass multiple lemmas (words in our context). Similarly, each lemma can be associated with multiple synsets. In our implementation, we calculate approximation metrics for every sense of lemmas matching our word-pairs and then select the highest similarity score.

One notable strength of WordNet databases lies in the high coverage of their vertical tree-based structure defining IS-A relationships. However, relatedness is better represented by horizontal relationships, which are cyclic and non-hierarchical (implemented via undirected graphs). Although WordNet defines some horizontal meronym/holonym relationships like PART-OF and SUBSTANCE-OF, they may not be sufficient in data coverage. For example, as illustrated in Figure 5.4, the words Turkey and Turkish, originating from the same root and being highly related, receive low similarity scores due to their distinct type paths (wup=0.23, path=0.07, lch=0.19). In the example, we observe two type paths for Turkish-as-a-language (Turkish → communication) and Turkey-as-a-country (Turkey → group) senses. If WordNet included a horizontal relationship like LANGUAGE-OF, relatedness algorithms such as hso (Hirst et al., 1998) could potentially provide better results (see no-relations line in the figure). For example, more comprehensive lexical resources such as Concept.Net (Speer et al., 2017) with 36 relationship types (e.g., Causes, MotivatedByGoal, UsedFor), seem to perform better in our pipeline. Taking into account the definitions of relatedness and similarity used in DSM studies (see §3.1), it can be argued that WordNet models similarity rather than relatedness due to their ability to define the proximity and distance between concepts using distinct type paths. However, our focus in this study is on relatedness.

In the AnlamVer study (Ercan and Yıldız, 2018), it was empirically demonstrated that relatedness and similarity are dependent variables. Specifically, the similar-unrelated sub-space within the Sim-Rel space contains zero items, indicating that if two concepts are already unrelated, they cannot be similar. However, a challenge arises in the other region of the space, where two concepts may exhibit relatedness but still display dissimilarity (with a similarity score less than 0.25). This situation poses a potential source of errors for WordNet algorithms modeling similarity, as exemplified by the case of Turkey and Turkish in Figure 5.4. To address this weakness, we introduce additional relatedness detection pipelines in the subsequent stages, leveraging type hierarchy and morphology.

Approximation Method Selection Experiments Among the methods we utilize, no single method has been reported in the literature to consistently outperform others. For instance, Agirre et al. (2009) presented Spearman correlation results of WordNet-based methods on the MC dataset, showing promising scores for wup (0.78), lch (0.79), res (0.81), lin (0.82), and jcn (0.83). Their distributional and hybrid approaches achieved even higher scores of up to 0.89 and 0.96, respectively. In our experiments, we obtained comparable results on the MC dataset with scores of wup (0.75), path (0.72), lch (0.72), res (0.73), lin (0.75), and jcn (0.82). Nevertheless, the MC dataset, consisting of merely 30 word-pairs with only frequent words, is relatively small, and it is arguably expected that correlation results would decrease as the dataset size increases. As demonstrated in Table 7.2 in Appendix, we tend to obtain lower results for larger datasets, such as 0.35 for WordSim353, 0.40 for MEN, and 0.49 for MTurk771.

Another study by Zhang et al. (2013) reports more varied results on the RG dataset, where wup and lch achieved the best scores of 0.78 and 0.79, while jcn performed the worst with a Spearman correlation of 0.58 (with res at 0.74 and lin at 0.62). Our results on the RG dataset range between 0.76 and 0.78. The same study also reports lower scores (max wup=0.38, min jcn=0.10) for the same experiments on Finnish datasets, Fin153 and Fin200, which can be attributed to the Finnish WordNet's lack of comprehensiveness. Despite covering a total of 24 methods on the RG dataset, the authors conclude that no single method consistently outperforms others on any dataset.

For the Turkish language, Sopaoglu and Ercan (2016) measured relatedness using three WordNet-based methods. We refer to their dataset as Sopaoglu (see Table 5.9), consisting of 101 word-pairs, 65 of which are translated from the original RG dataset. The scores were rated by 76 volunteer annotators, yielding an average inter-annotator score of 0.762. They reported the highest correlation (0.65) with their dataset using the wup method, while res and lch scored 0.59 and 0.55, respectively. In our experiments, wup yields the same correlation score of 0.65, while the path and lch algorithms achieve higher results. It should be noted that the Turkish WordNet used in our study is entirely

different (lexical entries, relationships, word coverage, implementation, etc.) from the one used by Sopaoglu and Ercan (2016).

Table 5.12 WordNet relatedness approximation experiments measured by Relatedness-classification and Word Relatedness tasks. Random (Rnd) and All-Rel are baseline classifiers. All-Related (All-Rel) is a dummy model which always predicts 'related'. Noun-to-noun and non-OOV word-pairs are included.

	Rnd	All-Rel	wup	path	lch	lin	jcn	res
English acc	0.50	**0.82**	0.78	0.50	**0.80**	0.63	0.23	0.63
English ρ	-	-	0.35	0.35	0.35	0.29	0.28	**0.38**
Turkish acc	0.50	0.66	**0.71**	0.53	0.69	-	-	-
Turkish ρ	-	-	**0.41**	0.36	0.36	-	-	-

Despite some hints in the literature regarding the leading performances of certain methods (wup, lch), we conclude that the methods included in this study do not consistently outperform others. We emphasize that a method's performance is heavily influenced by various resource parameters specific to each case, such as the evaluation dataset, WordNet implementation and data, the corpus used to feed IC, method implementations, and language. Considering the highly inflectional nature of the Turkish similarity dataset WordSimTr, which yields a 97% OOV rate on the WordNet database, we excluded it from our WordNet experiments. Throughout our WordNet approximation experiments, we only included noun-noun word-pairs and reported them as OOV.

Our objective is to identify the optimal binary relatedness classifier rather than focusing on ranking correlation. Therefore, we conducted our own experiments to **empirically determine the best-performing methods tailored to our specific task** and resources for both English and Turkish languages. We compared six WordNet methods to estimate word relatedness scores for word-pairs using conventional relatedness datasets (refer to Table 5.12). The results for aggregate word relatedness datasets consist of 6,170 word-pairs for English and 592 word-pairs for Turkish. These datasets are the combined versions of all relatedness datasets used in our study. To maintain the focus on relatedness, we

excluded the similarity datasets SimLex-999, WordSimTr, and AnlamVerSim, using the relatedness scores of AnlamVer word-pairs, which we refer to as AnlamVerRel. Following the threshold values t_x and t_y on the OSim-Rel space formulation, we labeled word-pairs as *unrelated* if their predicted relatedness values were lower than 2.5 and *related* if their values were greater than or equal to 2.5. To ensure comparability, we applied min-max normalization to the scores of some approximation measures (lch, jcn, res) that are not inherently normalized. Consequently, all values are converted to a scale ranging from 0 to 1. For Turkish, we limit our usage to three measures that do not require IC support because our WordNet implementation does not provide such support. After applying the threshold values on WordNet methods' predictions and ground truth scores of relatedness datasets, we report accuracy (acc) and ρ scores of each method in Table 5.12.

In addition, Table 7.2 in the Appendix presents per-dataset results for each approximation method, along with the full confusion matrix values of F_1, recall, and precision. Considering the **class imbalance** in the relatedness values of the ground-truth datasets, we also report F_1, precision, and recall measures. For English and Turkish, a significant proportion of word-pairs (16.9% and 32.10% respectively) are unrelated. Therefore, we include two benchmark columns to ensure a fair comparison of WordNet models, Random (i.e., Rnd) and All Relateds. The second benchmark column represents a *dummy* model that we refer to as *All Relateds* (i.e., All Rel), which statically predicts a binary related value for every sample. The All Related model achieves an accuracy score of 0.82, slightly outperforming the best approximation method (lch=0.80) in the English accuracy task. However, it cannot predict real values and fails to predict all negative (unrelated) samples.

In conclusion, our experiments for English demonstrate that lch performs the best in classifying relatedness with an accuracy of 0.80 and a F_1 score of 0.89, despite the res algorithm slightly outperforming lch on the Spearman correlation task (column ρ). For Turkish, wup yields the best results across measures, including ρ, accuracy, F_1, and recall. The datasets RareWords and

AnlamVer pose the most significant challenges in predicting word-pair orders, as reflected in their ρ values of 0.24 and 0.36, respectively, while performing similarly to other datasets in terms of accuracy. This aligns with our final word relatedness experiments (Table 6.7) for the RareWords dataset, which is considerably challenging, achieving a maximum ρ score of 0.43, while other relatedness datasets vary from 0.62 to 0.81. Based on the results, **we selected lch for English and wup for Turkish** as the WordNet approximation methods for detecting relatedness. Importantly, the winning algorithms for English do not utilize IC. This is appropriate as we intend to avoid evaluating corpus-driven DSM models using evaluation measures that are influenced by also corpus-driven factors.

5.3.2.4 Relatedness Filtering

At this stage, we aim to filter out all related word-pairs by utilizing all the resources we have gathered thus far and retain only the unrelated ones. Since we are automating the process of dataset creation, assessing the error margin for various stages, such as root detection, becomes challenging. To ensure the dataset's error kept to a minimum, we adopt a conservative stance, relying on the substantial size of the available word-pairs. From a strategic standpoint, our ultimate dataset emphasizes the **minimization of false negatives** over the maximization of word-pair quantity. As a result, our priority lies in mitigating false negatives (classified as unrelated but are actually related) rather than being concerned about false positives. In each sub-stage of the pipeline, if a positive (related) word-pair is found, it is removed, and the pipeline exits. Conversely, if a negative (unrelated) word-pair is found, the pipeline continues to the next stage. Table 5.13 displays the sub-stages of the pipeline.

5.3.2.5 Shared Root Detection

Within the Morphology stack (§4.6.1), the acquired roots undergo a matching process. If there exists at least one overlapping root among the roots,

we categorize the word-pair as related and consequently exclude it from the dataset.

Table 5.13 Stage 4: Relatedness filtering sub-stages.

ST	Relatedness Filtering	Filter Example	Reason to Filter-out
4.1	Shared Root Filter	airburst – airbus	MorphoLex detects shared root *_air*.
4.2	Semantic Filters		
	a) Relatedness Approx.	academy – academic	lch yields $0.31 \geq 0.25$
	b) Derivationally-Related	activity – activeness	Der-related-form record exists for the word-pair in WN.
	c) Type Hierarchy Match	anomalopidae – anomalops	*fish* from definition "a family of fish .." matches *fish* in types.
	d) Word Match	cosmogeny – cosmos	Synonym of cosmos *universe* matches a word in definition.
4.3	Categorical Filters		
	a) Type Blacklist	abelia – gambelia	One is *animal*, other one is *plant*. Both are in the blacklist.
	b) Type-pair Blacklist	acadian – akkadian	Abstract types *inhabitant* <-> *language* are in the blacklist.
	c) Common Meaningful A.	cyberart – cyberwar	*-cyber* adds its own meaning, it is in the affix blacklist.

5.3.2.6 Semantic Filters

In this section, we perform filtering by utilizing both the type hierarchy and text content matchings.

Relatedness Approximation Filter We know that the WordNet approximations achieve an 80% success rate in English (lch) and a 71% success rate in Turkish (wup) for relatedness detection (see Table 5.12). At this stage, we filter out all word-pairs that have been scored greater than 0.25 (relateds) according to the wup or lch algorithms. In the subsequent stages, we aim to compensate for this 20-30% error rate by eliminating false positive word-pairs.

Derivationally-Related Filter Following the assumption that *words derived from the same root are related* (see §4.5.2), we leverage the *derivationally-related-form* relations of words, which are already available in WordNet implementations. The derivationally-related-form entries between lemmas help reduce false positives to some extent by connecting certain words in a one-by-one manner (e.g., *abdication* – *abdicator*). However, especially in Turkish WordNet, we have observed that the data coverage of this relation is

quite limited. For example, in English WordNet, there are no defined relationships for activity other than *active* and *activeness*. However, there are numerous words derived from the root *_act*, such as *activism, reactivate, actor,* and *enact*. WordNet lacks the incorporation of the concept of *roots*, making it ineffective to associate every derivational pair with each other at the surface level.

Type Hierarchy Match Filter We retrieve the synonyms and definition texts of words from WordNet and then tokenize this information. The tokenization process augments the token set with root forms, leveraging the morphological stack of the language. We apply a minimum root length of 4 to avoid incorrectly matching stop-words. We subsequently check whether these tokens appear in the type hierarchy of the other word. When writing a word's definition within a sentence, there is a **high likelihood of using the type name that exists in the word's type hierarchy**. This tendency arises from the observation that a pattern similar to "{Target} IS-A {Type} with {Attributes} and {Relations}" is often followed during the process of writing definitions. Moreover, definition texts tend to provide a context that includes the closest neighbors of words, thereby **supporting the distributional hypothesis**. As shown in Figure 5.5, which provides examples of five filters in the pipeline, when defining the *anomalopidae* family, the definition text "a family of fish including: flashlight fishes" contains the word *fish*, representing the type of the object (Type Hier. Match in red). The concepts of the *anomalop* fish and its family name *anomalopidae*, which have not yet been defined by morphology and other filters in the pipeline, can be characterized based on the relatedness relationship identified by this filter. When matching tokens with the type hierarchy, we utilize type information up to a certain level of abstractness, which can be determined by a parameter (e.g., default is 75%). Depending on the length of type paths, we exclude matching for highly abstract concepts such as *entity, object, abstraction, communication.*

Word Match Filter In comparison to the prior filter, this filter differs by not inspecting the type hierarchy. Instead, it involves comparing a given word

and its possible synonyms with the tokenized definition of another word. As shown in Figure 5.5, the *anomalop* concept has a synonym, *flashlight fish*, which aligns with a token within the definition text of the other word. Throughout the orthographic matching process, all morphological and tokenization procedures employed in the previous filter are maintained.

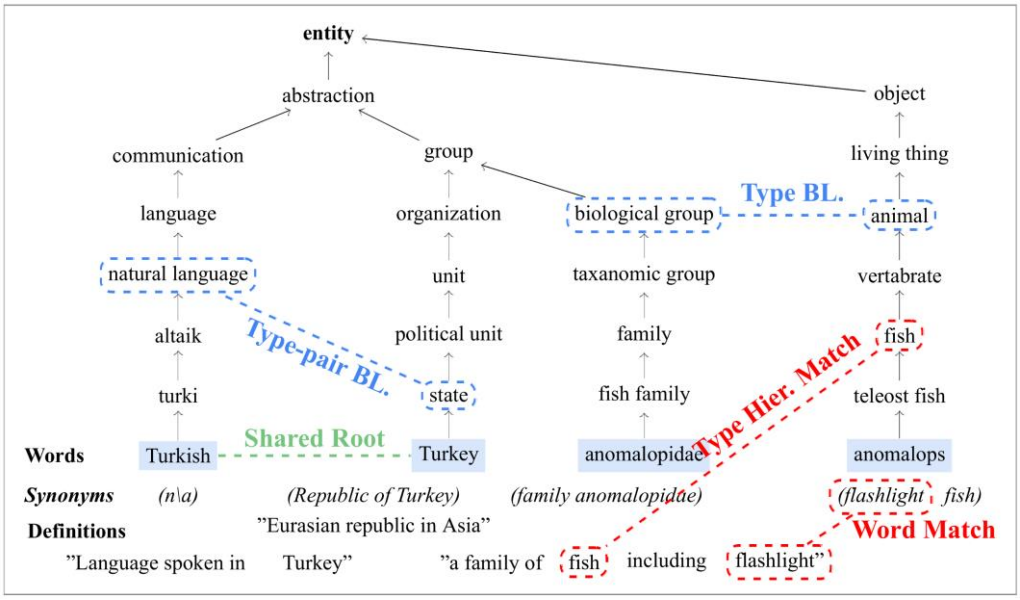

Figure 5.5 Simplified examples demonstrating filter types on WordNet type graph. Some concepts are omitted in the hierarchy for clarity. Definitions are shortened and changed slightly for clarity.

5.3.2.7 Categorical Filters

This stage entails researchers making specific definitions based on observations from their local experiments to address problematic areas. Accordingly, data samples from those identified areas are categorically eliminated. Considering the variations in language structures and the differences in WordNet implementations, these definitions are conducted separately for both languages. By taking into account the distinct language characteristics and unique WordNet resources, researchers ensure language-specific handling of data, leading to more accurate and reliable results for each language. Although

these filters are biased at the category selection level, they do not involve any selection intervention or bias at the word-pair instance level. The full list of categorical filters defined in the pipeline can be found in the shared source code.

Type Blacklist Filter In various domains such as plant, microorganism, and chemicals, specific terminologies with ancient roots, such as *antheridium*, *anomalopidae* and *helianthemum* are used. These specialized terms are not only scarce in our resources but also pose significant challenges in their morphological analysis for both English and Turkish languages. In contrast, English WordNet encompasses extensive taxonomies, including living species. However, discerning relatedness or similarity between such terms without resorting to internet resources is equally intricate for humans. In the realm of taxonomy, when a new insect species is discovered, it may be christened with a name derived from an ancient corn deity or the location of its discovery, as exemplified by *aegyptopithecus*. Consequently, this complexity renders the investigation of word and affix origins virtually impossible, especially for morphological decomposers. To address these challenges, a filtering mechanism has been implemented, comprising a blacklist of 14 types for English and 6 types for Turkish (e.g., *biological_group.n.01, animal.n.01, chemical.n.01*). These types are considerably abstract within the taxonomy. If a word-pair belongs to two types that are both present in the blacklist, the word-pair is excluded from consideration. As depicted in Figure 5.5, when *anomalopidae IS-A biological group* and *anomalop IS-AN animal*, we exclude it from the dataset. While applying this filter, the possibility of incorrectly eliminating numerous word-pairs as false positives is accepted.

Type-pair Blacklist Filter The main difference of this filter compared to the previous one is that it defines blacklists in type-pairs, not types. The domains listed in this blacklist don't necessarily have to be problematic as a whole. If both words in a word-pair match the types in a type-pair, we mark that word-pair as related and exclude it from further consideration. For instance, as seen in Figure 5.5, WordNet cannot model the obvious relatedness relationship between *Turkey* and *Turkish*. If the morphological analyzer fails to detect that these two

words share the same *_Turk* root, this pair might appear erroneously in the dataset. To resolve this issue, instead of defining instance-level relationships, we define **generic relatedness relationships by type-pairs at the abstract type level**. For example, when we state that there is a relatedness relationship between countries and languages, we automatically cover the instance *Romania* and *Romanian* as well. By intersecting vertical type graphs (four them in Figure 5.5) with 60 horizontal relatedness type-pairs for English and 42 for Turkish, we **bridge distinct type-graphs** and prevent hundreds of thousands of false matches of word-pairs. Some examples of these blacklisted type-pairs are: *inhabitant – language* (e.g., *acadian – akkadian*), *organic process – symptom* (e.g., *haematochezia – haematoma*), *body part – medical procedure (e.g., amygdala – amygdalotomy)*.

Common Meaningful Affixes As discussed by Bender (2013), the distinction between words and morphemes can be indistinct due to the dynamic nature of language change. In response to this, we have developed a categorical filter aimed at identifying affixes that convey actual meanings rather than modifying roots. Some affixes, such as *-cyber, -hyper*, and *+logy*, **convey their own meanings**, resembling constituents of compound units. To determine whether an affix is meaningful or not, we adopt the approach of randomly selecting a word and applying a potential *meaningful affix*. If, in doing so, every resulting unit (even made-up ones) feels related, we conclude that the unit should not be treated as an affix. This goes beyond the productivity of an affix. For example, consider the words *cyberart, cybersecurity, cyberwar, cybercrime, cybercafe*. If all of them feels related due to the presence of the *-cyber* affix, this situation is erroneous for our pipeline. To address this issue, we maintain a list of affixes that should not be treated as genuine affixes during the dataset construction phase. Consequently, if both words in a word-pair contain any of the aforementioned affixes simultaneously, we filter out that word-pair. Our list includes 15 affixes for English and 7 for Turkish (*-elektr, -nükleo, -karbo, +oloji, +grafi, +metri, +metre*) to account for their unique linguistic characteristics and usage patterns.

5.3.3 Reproducibility and Language Resources

We open-source the Python implementation of the dataset generation pipeline, named OSimUnr-Generator[30], to support the reproducibility of the methodology and facilitate its potential adaptation to additional languages. The repository is configured by default for English and the exact settings of the study but is designed to be extensible. The codebase is designed as a general NLP framework with features such as knowledge bases, orthographic similarity, word segmentation, and morphological modeling, with extensibility and testability in mind. We encourage researchers to fork the codebase and follow the documentation to add new languages or modify parameters. For adding new integrations and algorithms, the it includes comprehensive code-level documentation as well as unit and integration tests to assist in the process.

5.3.3.1 Assumptions and Parameters

Based on the OSIM-REL space definition (Fig. 3.4, Eqs. 3.1 and 3.2) and the morphological assumptions of the study, the generator pipeline defines some default threshold values as parameters for researchers to customize. For example, the t_x 'unrelatedness' and 'highly related' threshold levels are defined arbitrarily as 2.5 on the 0-10 scale system in order to symmetrically divide the semantic x-axis. Similarly, the t_y axis is defined in the same manner to represent the level of orthographic similarity, which defines the Q3 and Q4 sub-spaces. The generator pipeline starts accepting these threshold values as parameters regarding relatedness and orthographic space of the systems. It uses a 0-1 scale system. Some essential API parameter definition shown in Table 5.14.

[30] http://gokhanercan.com/OSimUnr-Generator

Table 5.14 Essential API parameters and descriptions

Parameter Name	Description
`wordPosFilters`	Defines the part-of-speech (POS) tags that the word-pool should use. Default is POSTypes.NOUN.
`minOrthographicSimQ3`	Defines the lower limit of the Q3 orthographic space. The upper limit is minOrthographicSimQ4. Default is 0.50.
`minOrthographicSimQ4`	Defines the lower limit of the Q4 orthographic space. The upper limit is 1 by default. Default is 0.75.
`maxRelatedness`	Defines the maximum level of 'unrelatedness' of word-pairs on a scale of 0 to 1. Default is 0.25.

5.3.3.2 Extensibility

To provide extensibility, OSimUnr-Generator supports the Provider design pattern, allowing researchers to easily modify and extend the pipeline with additional algorithms and resources without altering the core dataset generation behavior. Below is a code snippet to initiate the generation process:

```
lang = LinguisticContext.BuildEnglishContext()
orthoAlg = EditDistance()
pipe = EnglishPipeline(lang, orthoAlg)
pipe.GenerateDataset(POS.Noun,0.50,0.75,None,0.25)
```

EnglishPipeline is the default concrete provider implements the following factory methods of the PipelineProviderBase class (Fig. 5.6), organized into three groups; morphological resources, semantic resources, and filtering data. Filtering data methods allow manual definition of filters, as explained in the Categorical Filters section (5.3.2.4). Although the EnglishPipeline implementation heavily relies on NLTK WordNet for the word pool, semantic relatedness approximation, and shared root detection, the system depends on the IWordSource, IWordNet, and IRootDetector abstractions. This design enables researchers to implement alternative solutions easily, as achieved in this study, where the Turkish pipeline employs an entirely different implementation by consuming Java services. The MorphoLex dependency is used as a minor part

of the dependencies, in contrast to WordNet, which serves as a more central component.

```
class PipelineProviderBase(ABC):

  def __init__(self, ctx, osimAlgorithm):
    self.Context:LinguisticContext = ctx
    self.OSimAlgorithm:IWordSimilarity = osimAlgorithm

  #Morphological Resources
  def CreateRootDetector(self) -> IRootDetector:pass
  def CreateFastRootDetector(self) ->IRootDetector:pass
  def CreateTokenizer(self) -> ITokenizer: pass

  #Semantic Resources
  def CreateWordNet(self)->IWordNet: pass
  def CreateWordSource(self) -> IWordSource: pass
  def CreateWordNetSimAlgorithm(self)->WordNetSimilarity

  #Filtering Data
  def CreateBlacklistedConceptsFilterer(self,pos)
  def CreateConceptPairFilterer(self, pos: POSTypes)
  def CreateDefinitionBasedRelatednessClassifier()
  def CreateDerivationallyRelatedClassifier()
```

Figure 5.6 Simplified abstract PipelineProviderBase class

5.3.3.3 Availability of Language Resources

Irrespective of the ease of technical extensibility, the dataset generation and modeling phases are inherently dependent on annotated data, primarily NLTK WordNet (Ehsani et al., 2018) and MorphoLex (Sánchez-Gutiérrez et al., 2018). In terms of quantity, the initial word pool sizes, prior to POS and punctuation processing, are 147,306 for English and 80,942 for Turkish (Table 5.11). Similarly, the primary components of the English[31] and Turkish morphology stacks, MorphoLex and MorphoLex Turkish, contain 70,000 and 48,472 morphological decompositions respectively, all annotated by linguists. Regarding quality and structure, for both languages, we employed fully derivational morphology, modeling nearly all roots and affixes available in these

[31] English morphology stack EnglishRootDetectionStack.py, is publicly available at https://github.com/gokhanercan/OSimUnr

languages (tr: 405 affixes, en: 467 affixes). Due to the highly productive agglutinative morphology of the Turkish language, characterized by extensive derivation and inflection, we utilized a finite-state transducer library, the Turkish Morphological Analyzer (Yıldız et al., 2019), which was customized for this study to support derivational morphology with an atomic roots lexicon. As discussed in Section 3.2.2.2, we argue that as the synthesis level and orthographic transparency increase, the effectiveness of using a finite-state machine for modeling a language to reduce noise also tends to improve. These resources were deliberately designed to ensure high quality, thereby enhancing both the dataset and the modeling process. This approach reduces the number of false negative word-pairs in the dataset and allows for effective modeling of the possible roots and affixes.

Table 5.15 Resource availability for new language adaptation

Resource Availability	Voc#	Forms#	Rel.Appr.	Lexicon	#	Languages (ISO 639-3)
Fully Implemented	55,350	38,340	NLTK WN	MorphoLex-fr	3	**eng**, ***tur***, **fra**
Requires UniMorph Impl.	20,000	50,000	NLTK WN	UniMorph 4	8	**cat**, ***fin***, **ita**, nld, **pol**, **por**, slv, **spa**
	20,000	10,000	NLTK WN	UniMorph 4	3	*eus*, glg, ind
	3,000	10,000	NLTK WN	UniMorph 4	10	ara, bul, dan, ell, fas, heb, nno, nob, sqi, **swe**
Requires Two Impls.	20,000	50,000	CN5.5	UniMorph 4	17	fro, **ger**, gle, hin, **hbs**, ***hun***, hye, isl, *kat*, **lat**, lav, mkd, ron, **rus**, slk, sme, xcl
	20,000	10,000	CN5.5	UniMorph 4	8	ast, bel, **ces**, *est*, grc, *kaz*, lit, ukr
	3,000	10,000	CN5.5	UniMorph 4	19	*ady*, afr, ang, cym, dsb, fao, frm, *guj*, *nav*, oci, osx, *que*, *sah*, san, syc, urd, *uzb*, vec, *vep*
	0	0	CN5.5	UniMorph 4	47	amh, *arn*, *aze*, ben, bod, bre, *ceb*, chu, cor, crh, csb, *dak*, *dje*, fry, frr, fur, gla, glv, gmh, gml, goh, gsw, *hil*, kan, *kal*, *kbd*, *kir*, *kjh*, *krl*, *lin*, liv, lld, *mlg*, mlt, nap, pus, sga, *sot*, *tat*, *tel*, tgk, *tuk*, *tyv*, *uig*, yid, *vot*, xno

Rows are ordered by resource availability and readiness level, from highest to lowest. The Voc# columns represent the minimum vocabulary size for the Relatedness Approximation Resource. The Forms# columns represent the minimum number of morphological forms (inflectional and/or derivational) available in the Lexicon database. The NLTK WordNet and MorphoLex resources are already implemented in the OSimUnr-Generator Python library. Agglutinative languages are italicized. Languages with inflectional segmentation data in the Lexicon are bold, and those with derivational data are underlined.

However, such resource availability is not feasible for all languages. To the best of our knowledge, no universal expert-annotated derivational segmentation database or morphological analyzer currently exists that supports

decomposition into atomic units and multiple roots. Even though there are many language-specific resources specialized for individual languages (e.g., several advanced Turkish morphological analyzers for Turkish (Yıldız et al., 2019)), the number of universal databases and analyzers remains very limited. It appears that current resource landscape aligns with Bender's statement (Bender, 2013): *"...while general methodologies for building morphological analyzers can be applied across languages, there will always be "language-specific work to carry out, either in creating rule sets or in annotating data..."*. Given the high cost of integrating existing language resources, reusing implementations such as WordNet and MorphoLex is essential for adapting to new languages and ensuring the reproducibility of this study. In Table 5.15, we present statistics on the availability of resources and their adaptability to new languages, based on the hypothetical inclusion of two additional universal resources in the pipeline of similar research.

Off-the-Shelf Resource Implementations The first row of the Table 5.15 highlights French (fra) as the only fully implementation-ready resource, aside from Turkish and English, as it has a MorphoLex-fr (Mailhot et al., 2020) variant and is supported by WordNet. MorphoLex-fr contains 38,840 French word decompositions in the same format, and the WordNet synset graph includes 55,350 French word lemmas (i.e., vocabulary). To our knowledge, MorphoLex variants are currently limited to English, Turkish, and French. In total, NLTK WordNet provides a graph hierarchy for 29 languages in the shared OMW 1.4 format, as provided by the Open Multilingual WordNet (OMW) project[32], 18 of which contain more than 20,000 words. There is also an experimental version in which the authors utilize the newer OMW 2.0 format, expanding the coverage to 40 languages (Bond et al., 2020).

Table 5.15 presents resource availability in descending order, grouping languages by vocabulary size into categories such as more than 20,000, more than 3,000, and fewer than 3,000 words. Similarly, the number of inflectional

[32] https://omwn.org

and/or derivational forms in the derivational database is grouped into categories of more than 50,000, more than 10,000, and fewer than 10,000 forms. These thresholds are intentionally set to ensure balanced dataset splits and were determined based on practical considerations and empirical observations of availability. Table also lists the languages that fall into these groups, based on data from the two new universal resource databases, ConceptNet and UniMorph.

UniMorph 4 UniMorph (Batsuren et al., 2022), created through a collaborative effort of numerous linguists, began as an inflection database featuring 23 semantic tags and 212 feature tags. It includes automatic extraction from various resources such as Wiktionary and covers 182 languages, including 30 endangered ones listed by United Nations Educational, Scientific and Cultural Organization. The database comprises 122 million inflections and 769,000 derivations and features a language-independent schema, making it highly adaptable to various linguistic research applications. The most valuable components of the dataset for research like ours, segmentation and derivational resources, are unfortunately limited to 30 languages for derivations and 16 languages for inflectional segmentation. Assessing the quality of suffixation is challenging, but since it is not originally a segmentation database, we cannot claim it is comparable to MorphoLex for most languages due to the synthetically generated nature of the derivational dataset, its lack of atomic roots, and the absence of an affixation-per-entry structure. For example, UniMorph's inflectional segmentation record for the word impracticality is "impracticality" since it has no inflections, with "impractical-ity" as its derivational record, while MorphoLex's segmentation is "<im<{(pract)>ic{>>}al>}>ity>". With our shallow suffixation analyzer implementation in the pipeline, it is possible to cover a greater number of surface realizations using MorphoLex, with its 70,000 records, compared to the UniGraph English dataset, which contains 652,477 inflectional segmentation records and 225,131 derivation records.

Another universal resource that can be used as a segmented lexicon, UniSegments (Žabokrtsky` et al., 2022) is accompanied by a detailed paper that surveys 17 language-specific derivational databases across 32 languages. It

introduces a harmonized scheme for segmentation representation, converting and standardizing the data from the studied resources into a single, unified format. Similar to UniMorph, UniSegments extends MorphyNet (Batsuren et al., 2021), a multilingual morphological database with 519,000 derivational and 10.1 million inflectional entries.

ConceptNet 5.5 For relatedness approximation, one alternative universal resource is ConceptNet (Speer et al., 2017), an open multilingual knowledge graph representing 304 languages[33], each with at least 300 words. ConceptNet includes 10 highly represented languages that provide full API features, encompassing 9.5 million words, and 68 common languages, each with at least 10,000 words. It is derived or extracted from various sources, including Wiktionary, Open Mind Common Sense, WordNet OMW, OpenCyc, DBPedia, and various games designed in a "games with a purpose" fashion (Von Ahn, 2006). ConceptNet supports 36 relationship types, including *RelatedTo, CapableOf, Causes, Entails, FormOf, HasA, UsedFor*, and others, most of which can be interpreted as modeling relatedness rather than similarity. It also includes the *EtymologicallyRelatedTo* and *EtymologicallyDerivedFrom* relationships, which are equivalent to the *derivationally-related-form* relationship in WordNet and are utilized in the shared root detection stacks. The resource is fully downloadable or can be accessed via a managed API with request limits. Additionally, it offers an endpoint to calculate the relatedness score between two given words. The languages listed in the "Requires Two Implementations" section of Table 5.15 are grouped based on ConceptNet vocabulary size categories and the availability of lexical resources, filtered to include only those that satisfy both criteria.

[33] https://github.com/commonsense/conceptnet5/wiki/Languages

CHAPTER 6

6. EXPERIMENTS

6.1 EXPERIMENT SETUP

6.1.1 Experiments

6.1.1.1 Experiment 1 - Subword-level Unrelatedness Identification

We conducted four types of experiments for the evaluation. Our first experiment type focuses on testing the distinguishing capability of subword-level models. We achieve this through a task we propose as *unrelatedness-identification*, which evaluates discrete and continuous (acc and mae) errors of model estimations using the OSimUnr dataset we built (see Table 6.4 for experiment results). All sub-datasets of OSimUnr in all dimensions—Q3 and Q4 groups generated by both orthographic similarity measures over_ft23 and editsim—are included in these experiments. In Experiment 1, only subword-level models (e.g., FT-*) are utilized, expecting the models to respond to OOV word-pair queries as well. This constitutes a one-class classification task, as it includes only the positive (unrelated) side of the classification. Consequently, we report accuracy derived solely from the confusion matrix.

6.1.1.2 Experiment 2 - Word Relatedness (Subword-level)

The second experiment type focuses on controlling the relative performance of semantic models. It takes the form of a traditional *word similarity task* that is evaluated using Spearman ranking correlation ρ of word-pair estimations on popular datasets (see Tables 6.7 and 6.8). This task ensures that we do not compromise performance on an existing *relative task* while improving our primary objective of distinguishing ability (Table 6.4). The result score ρ of this task is plotted on the y-axis of our proposed Semantic Clarity

Space, while the primary objective is represented on the x-axis (see Figure 2.4 and 7.1).

6.1.1.3 Experiment 3 - Word-level Unrelatedness Identification

The third experiment aims to demonstrate that word-level semantic models, such as Word2Vec, are capable of distinguishing words from each other, unlike n-gram-segmented FastText models, which suffer from this limitation (Table 6.6). If n-grams are the root cause of the noisy spaces, word-level models should not have any noise and consequently should not struggle with distinguishing unrelated words from each other. In this type of word-level experiment, we exclude OOV word-pairs to ensure comparability between word-level and subword-level models.

6.1.1.4 Experiment 4 – Relatedness Classification

Our final experiment aims to evaluate the models' ability to detect the negative (related) side of the relatedness dimension. Since the OSimUnr datasets (Experiments 1 and 3) exclusively represent the positive (unrelated) side of the ground truth data, we report only the accuracy of the models' predictions for positive labels because other metrics such as F_1 score, recall, or precision are uninformative when false positives (FP) and true negatives (TN) are zero. Consequently, Experiments 1 and 3 do not include these metrics.

To extend this evaluation, we created two additional sub-datasets containing discrete labels for both related and unrelated word-pairs, allowing for a more comprehensive assessment of the models' binary classification performance. These datasets are imbalanced and heavily weighted toward the related side, creating a challenging evaluation scenario for models that typically assign low relatedness scores to word-pairs.

WordSims The first sub-dataset, WordSims[34], is a combined version of all relatedness datasets used in this study (Table 5.9). It includes 6,170 word-

[34] https://github.com/gokhanercan/OSimUnr/blob/master/others/ WordSims-REL-EN.csv

pairs for English and 592 word-pairs for Turkish, all scored by human annotators and normalized to the same 0-1 scale for consistency. This dataset is also used for Spearman evaluation of the word relatedness task in Experiments 2a and 2b (Table 6.7,6.8). In this experiment, and following the study's relatedness assumption, the dataset is treated as a two-class related/unrelated dataset, with records considered related if their scores are greater than 0.25. The balance of related and unrelated records is as follows: For English, 82% of the records are related, while 18% are unrelated. For Turkish, 66% of the records are related, while 34% are unrelated.

OSimBinary The second sub-dataset, OSimBinary[35], was created using the generator pipeline (Stage 4 in Table 5.11). Unlike the other OSimUnr sub-datasets, it includes both related and unrelated word-pairs with the isrelated label by retaining the related-detected word-pairs instead of filtering them out. Only the blacklisting substages, such as Type Blacklist, and Type-pair Blacklist (Table 5.13), remain in effect, excluding specific word-pairs from the dataset. We selected the dataset from the editsim Q4 pool (EN: 54,574 word-pairs, TR: 30,905) to make it more challenging for models sensitive to orthographic similarity. After applying blacklisting, the dataset was reduced to 53,771 English word-pairs and 30,689 Turkish word-pairs. Unlike the WordSims dataset, relatedness values in OSimBinary are not human-annotated ground truth but are instead derived from WordNet-relatedness approximations and root detection assumptions. The class imbalance is even more pronounced toward relatedness, with 95% of word pairs in English and 94% in Turkish classified as related. This distribution stems from the WordNet database and relatedness approximation algorithms. When selecting a random word pair from the WordNet word pool, the probability of it being unrelated is approximately 5%, even though all of these word pairs are orthographically highly similar. Another difference from the WordSims dataset is that these word pairs tend to be infrequent due to the presence of many terminological and proper nouns (e.g., *acrimony, Aigina*),

[35] https://github.com/gokhanercan/OSimUnr/blob/master/ S3-OSimBinaryQ4-editsim-EN.csv

whereas the WordSims dataset consists of manually curated pairs and is biased toward frequent words, as explained in Section 5.3.1.2. These characteristics make this sub-dataset the most challenging element in our experiments.

6.1.2 Measures

Aside from the traditional Spearman ranking correlation p measure of the word-relatedness task (Table 6.7), we also utilize the following measures:

6.1.2.1 Accuracy (acc)

The primary performance measurement of the study is the overall accuracy (i.e., acc) of the unrelatedness-identification and relatedness-classification tasks. We achieve binary results by applying a relatedness threshold value using the IsUnrelatedm(w1, w2) function to continuous model (m) predictions we get from the Relm(w1, w2) function (Equations 6.1 and 6.2). As explained in Section 5.3.2.4, the ground truth labels of this task are unrelated OSimUnr word-pairs which are achieved by applying the same threshold function IsUnrelatedwn(w1, w2) to normalized WordNet (wn) relatedness approximations (Relwn(w1, w2)). Although OSimUnr ground-truth relatedness approximations are normalized between 0 and 1, we normalize all model predictions between 0 and 10 before converting them into binaries. This is done to align with the 0-10 scale of the OSim-Rel space and threshold variables. We compute the final accuracy by dividing true predictions (TP and TN) by total number of predictions (Equation 6.3).

$$IsUnrelated_m(w_1, w_2) = Rel_m(w_1, w_2) < t_x \tag{6.1}$$

$$TruePrediction_m(w_1, w_2) = IsUnrelated_m(w_1, w_2) = IsUnrelated_{wn}(w_1, w_2) \tag{6.2}$$

$$acc = (TP + TN)/(TP + FP + TN + FN) \tag{6.3}$$

$$pre = TP/(TP + FP), \;\; rec = TP/(TP + FN) \tag{6.4}$$

$$F_1 = 2 \cdot pre \cdot rec/(pre + rec) \tag{6.5}$$

$$Specificity = TN/(TN + FP) \tag{6.6}$$

6.1.2.2 Recall, Precision and F1 Scores

Considering the imbalance of the datasets, and the fact that the models also produce imbalanced predictions, we evaluate the WordSims and OSimBinary datasets in Experiment 4 (Table 6.5) using the standard precision, recall, and F_1 measures, as defined in Eqs. 6.4,6.5. The importance of precision or recall varies depending on the upstream task utilizing the classifier. Since we do not prioritize one over the other, we adopt F_1 as a balanced metric that considers both measures equally. In applications requiring high recall and/or high specificity (Eq. 6.6), such as a text editor detecting unexpected word instances like misspellings or the use of irrelevant words in context, the system should aim to exhaustively identify all possible related or unrelated occurrences. For instance, for the erroneous sentence "Souffle the dataset for analysis," the system should determine that the word-pairs *souffle – shuffle* and *dataset – souffle* are unrelated, while *shuffle – dataset* is related, to ensure the error is not missed. Conversely, in applications where high precision is prioritized and lower recall is acceptable—such as automatically generating multiple-choice exam questions (e.g., identifying irrelevant word usage or selecting the most irrelevant word)—the classifier's decisions can directly correspond to the correct answers for the test.

6.1.2.3 Mean absolute error (err)

Since our main tasks are to distinguish two unrelated concepts from each other, we inevitably applied **hard thresholding** using the "IsUnrelated function" (Equation 6.1) while converting continuous semantic model predictions to binaries. The accuracy measure is arguably prone to false classifications due to the arbitrary threshold value t_x we choose and the varying distributions of model predictions. The assumption that "all word-pairs greater than 2.5 are related" might be error-prone because unlike our presumptions, our empirical results show that **FastText DSMs do not generate "well-distributed" data predictions**. For example, Figure 6.1 shows that the distributions of relatedness

can differ significantly in the bell shape's curve and x-axis offset when the only varying parameter is the objective, SkipGram or CBOW. In both W2V(SG) histograms, the variance of predictions (oranges) is very low, and the unrelated (<0.25) and highly-related (>0.75) areas are almost not represented. In contrast, the variance of CBOW predictions (W2V) is higher and the unrelated space is fairly well represented. Therefore, it is almost impossible to target the *unrelated* area of the space with the SkipGram objective. It is important to note that the distributions in Figure 6.1 represent word-level semantic spaces, excluding the noise caused by subword-level segmentation methods. Considering this potential weakness of hard thresholding, we employ a second supporting measure: the mean absolute error (i.e., mae or err), which quantifies **continuous error** between the model prediction and the ground-truth value y in the dataset (Equation 6.7). Although this measure has not been widely used in DSM evaluation, it still holds value as it provides an intrinsic benchmark for different model configurations. For example, in their study focused on measuring compositionality, Lazaridou et al. (2013), utilized a similar measure called 'mean similarity of vectors' as an intrinsic evaluation method. They reported the mean error between composed vectors and corpus-extracted derived-form vectors to benchmark various composition methods. We include the err as an additional measure alongside the Spearman ranking correlation p in conventional wordsim dataset experiments (Table 6.7).

$$err(w_1, w_2) = mae(w_1, w_2) = |y - Rel_m(w_1, w_2)| \tag{6.7}$$

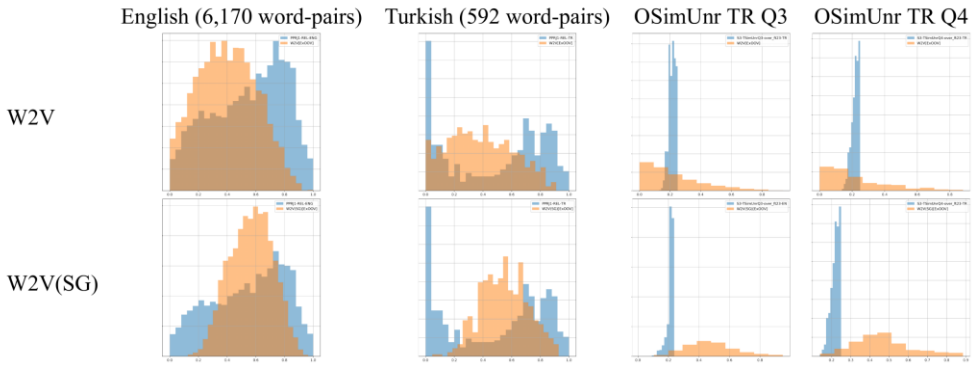

Figure 6.1 The histogram shows how relatedness distributes in a word-level SkipGram semantic space. Blue areas represent ground-truth word relatedness scores, while orange areas represent model predictions. Browns areas indicate overlapping regions.

6.1.3 Corpora

We followed the same corpus pipeline steps for both languages: including combining, preprocessing, building frequency statistics, and morphological annotation. Initially, all corpora underwent cleansing of punctuation marks and extra whitespaces, tokenization, conversion to lowercase, and shuffling of sentence order. In contrast to the English corpus, publicly available corpora for Turkish are limited in size. To overcome this limitation, we combined multiple Turkish corpora into a single corpus with the aim of approaching the scale of the English corpus. In both languages, the final corpora exhibited vocabulary sizes of over five million unique tokens (en=5.5M, tr=5.2M). As shown in Table 6.1, the vocabulary size (number of unique tokens) and the number of tokens in our final corpora are proportionate (en=1.5B, tr=1.24B).

Given the nature of the encyclopedia domain, sentences in our English corpora tend to be longer, more informative, and contain a higher number of unique tokens compared to those in other domains. Our English corpus, PolyglotWikiEN13 (Al-Rfou et al., 2013), comprises 70 million Wikipedia sentences with 5.5 million unique tokens. It has an average of 21.5 tokens per sentence. In contrast, our base corpus for Turkish, BounWebCorpus (Sak et al., 2011), has an average of 12 tokens per sentence. Despite adding the

OpenSubtitles2018 Corpus (Lison et al., 2018), which consists of a significant number of sentences, to our Turkish corpus, we anticipated that the limited diversity and informativeness of the data (4.6 tokens per sentence) would still be insufficient. To address this, we added two Wikipedia-based corpora, trwiki-67 (Safaya et al., 2022) and PolyglotWikiTR13 (Al-Rfou et al., 2013). Although they were compiled using different extraction techniques in different years, there is a possibility of overlap between them.

Table 6.1 Corpora utilized in experiments. VocSize: Vocabulary size (unique tokens), Sent: Number of sentences, M: millions, B: billions. The Turkish corpus is a union of four separate corpora.

Corpus	Source(s)	Domain	Sent.	VocSize	Tokens
English	PolyglotWikiEN13 (Al-Rfou et al., 2013)	Encyclopedia	70M	5.5M	1.51B
Turkish	BounWebCorpus (Sak et al., 2011) ∪ OpenSubtitles2018 (Lison et al., 2018) ∪ PolyglotWikiTR13 (Al-Rfou et al., 2013) ∪ trwiki-67 (Safaya et al., 2022)	News, Websites Movie Subtitles Encyclopedia Encyclopedia	189M	5.2M	1.24B

6.1.4 Model Configurations

In our experiments, we primarily used the Continuous Bag-of-Words (CBOW) objective of the FastText model (FT-*) with its default hyperparameter settings, including dimensions (dim) of 100, window size (win) of 5, n-gram range of [3-6], learning rate of 0.025, hash bucket size of 2,000,000, and so on. To enhance sensitivity to OOV and rare-word scenarios, we adjusted the minimum word frequency threshold from the default value of 5 to 0. For the purpose of conducting distribution comparisons, we separately experimented with the SkipGram and CBOW objective parameters in each experiment. To facilitate word-level benchmarking with consistent objectives, we included the Word2Vec (W2V) model in our experiments, using its default hyperparameter settings (win:5, dim:10, negative sampling:5, among others).

To maintain consistency across different configurations, we employed various word-segmentation methods, ranging from trivial to morphologically complex approaches. When character n-gram-based segmentation (CG) was utilized, we employed the [3-6]gram setting. However, for morphological units, we used the (1-1)gram setting, which cancels out the n-gramming algorithm and **represents each morpheme with a single vector**. Morphemes with the same form but different types (prefix, root, suffix) were represented by separate vectors. For instance, the form a exists in MorphoLex in all types (*-a, _a, +a*). Thus, we represent the prefix with the vector v_{-a}, the root with v_{a}, and the suffix with v_{+a}. Since FastText's objectives represent words in a bag-of-units fashion, the subword unit **orders are ignored** in our configurations. Therefore, semantically different instances such as *_göz+lHk+CH+lAr* (opticians) and *_göz+**CH**+**lHk**+lAr* (lookouts) were considered equivalent in the models, which is not an uncommon case in Turkish. Additionally, in line with FastText's default practice of utilizing bag-of-subwords, we also added an extra vector for the surface form of the words (*gözlükçüler*) for all segmentations.

Throughout our tests, we examined the impact of various hyperparameter variations on the performance of FT-SG and FT-CB models in the OSimUnr task. Notably, we explored variations in iteration count, minimum word frequency threshold of 5, and different char-gram configurations, such as CG[1-2], CG[2-3], and CG[1-4]. These hyperparameter variations did not yield significantly different results. Despite these initial observations, we acknowledge that a more systematic hyperparameter investigation may be warranted to optimize models for distinguishing-ability purposes.

In morphological configurations, there is no need to utilize the *hashing-trick*, which FastText employs for performance and memory optimization purposes. As described in the book by Bhattacharjee (2018), the hashing-trick involves hashing subword (char-gram) vectors in models into a limited space, typically 2,000,000, while disregarding collisions. FastText's hashing-trick implementation relies on the assumption that frequent subwords, following Zipf's Law, will occupy the hash space before rare-words, and hashing collisions

will occur among insignificant units. However, in our morphological segmentations, we have a bounded number of morphological units, making such an application unnecessary. Given that our annotated corpora contain information about the total number of roots, affixes, and words, we can determine the unit size of matrices in advance. For example, in the English corpus, aggregating 5.5 million unique surface words with 15,477 roots, 144 prefixes, and 278 suffixes allows us to determine the total unit size of the matrix. In this specific scenario, morphological models **exhibit superior efficiency in both memory and computational requirements** compared to char-gram models.

6.1.5 Word Segmentations

In our experiments, the main differentiating factor is the word segmentation method, as we use the Continuous Bag-of-Words (CBOW) and SkipGram (SG) objectives of the FastText (FT) and Word2Vec (W2V) models with fixed hyperparameter settings. As shown in Table 6.2, we use a total of four different word segmentation methods in our experiments. Our model configuration naming convention follows the format ”Model-Segmentation(Objective).” For example, FT-MR(SG) refers to the FastText model with root-only morphology trained using the SkipGram objective. When we do not specify the objective, the default objective we use is CBOW.

Table 6.2 Word segmentations by examples. The [3-6] and (1-1) notations are default grammings of segmentations. Notation: gram *-prefix _root _segment +suffix*

Word Segmentation		**Turkish**	**English**
	Surface form	*gözlükçü*	*unselfconsciousness*
CG	Char-gram [3-6]	*<gö <göz ... kçü>çü>*	*<un <uns ... ess>ss>*
HYP	Hyphenation (1-1)	*_göz_lük_çü*	*_un_self_con_scious_ness*
M	Morphological (1,1)	*_göz+lHk+CH*	*-un_self_conscious+ness*
MR	Morphological roots (1,1)	*_göz*	*_self_conscious*

6.1.5.1 Char-gram (CG)

In this configuration, FastText's default n-gramming algorithm CG[3-6] is used. The start and end characters (<,>) that differentiate the beginning and ending n-grams from the middle n-grams are also included in the segmentation. For example, <*gl* represents an n-gram with the starting character and *ng*> represents an n-gram with the ending character (see the example in Table 3.1).

6.1.5.2 Hyphenation (HYP)

We incorporate hyphenation (HYP), also known as syllabification, as an alternative segmentation method due to its position between two extremes: the meaningless character n-grams and morphemes that carry significant morphological meaning. While syllabification rules vary across languages, they are not as arbitrary as individual letters, suggesting that they may offer a middle ground in terms of the *distinguishing words* task performance. For English hyphenation, we utilize the pyphen[36] library, which relies on Hunspell hyphenation dictionaries. This library provides comprehensive hyphenation rules for English words, enabling accurate segmentation into syllables. LibreOffice[37] uses the Pyphen library to provide hyphenation support for 39 languages.

In the case of Turkish, syllabification follows relatively straightforward principles with the exception of loan words. The basic rules are: i) "all syllables contain one vowel", ii) "a vowel cannot be the first item in a syllable unless it is at the beginning of a word", iii) "a syllable cannot begin with two consonants, except at the beginning of loan words," and iv) "at the end of a line, a word can be divided at any syllable boundary" (Göksel and Kerslake, 2004). For Turkish syllabification, we have developed our own Java implementation that does not rely on any lexicon or training data. In both languages, hyphens are considered

[36] https://pyphen.org
[37] https://www.libreoffice.org

relatively meaningful units. Consequently, we adopt the (1-1) configuration settings for hyphenation in our experiments.

6.1.5.3 Morphological (M)

Morphological segmentation in this study incorporates all the obtained morphological units, including multiple roots, prefixes, and suffixes for both languages (e.g., *-un_self_conscious+ness*). These meaningful units are modeled in a bag-of-morphemes fashion, as described in the Morphology section. The configuration for morphological segmentation is different from char-gramming in that it uses (1,1) gramming settings, meaning that we do not add start and end morphemes. Each morpheme has only one vector representation, regardless of its position in the word or its co-occurrence with other affixes. This assumption implies that **each morpheme always has a single meaning**, which may not always hold true in all cases. For example, in Turkish, the word *gözlükçülük* consists of two instances of the *+lHk* derivational suffix. In the first instance, it transforms the root word *göz* (eye) into *gözlük* (glasses), while in the second instance, it changes *gözlükçü* (optician) into *gözlükçülük* (occupation of being an optician). Since both instances of *+lHk* are represented by the same v_{+lHk} vector, these differences cannot be modeled.

Another aspect of this study is that we fully support derivational and inflectional affixes without making any distinction. Therefore, we learn separate vectors for tense markers (tr: *+DH, +Hyor*; en: *+ed, +ing*) and plural markers (tr: *+lAr*; en: *+s*), even though these affixes do not add meaning to the words they attach to (see the assumption in §4.5.5). Since our main evaluation task focuses on word-pair comparison and does not involve sentence context, the distinction between derivation and inflectional affixes does not make a significant difference, as there are few instances of inflected words in the WordNet lexical word-pools we use for word-pair selection (e.g., *_doom+ed* or *_dress+ing*). However, a more crucial aspect that affects model performance is the **treatment of productive derivational affixes**. In both languages, affixes such as *+tion, +ness, +CH, +lHk* can be added to any word and systematically

alter its meaning to some extent (e.g., *_lazy+ness*) (assumption 4.5.4). Since these affixes can be applied to all words in any context, their inclusion in a simple bag-of-units model may cause more problems than benefits. Taking into account the types, order, and relationships of these morphological segments along with other morphological information such as part-of-speech, affix types, and morphological tags, represents a more advanced modeling objective that we leave for future studies.

6.1.5.4 Morphological Roots (MR)

In the Morphological Roots (MR) segmentation, we simplify the morphological model (M) by reducing words to their root morphemes only. This segmentation specifically excludes all types of affixes within the model space. For example, in Turkish, the word *gözlükçü* is reduced only to the root *_göz* (Equation 6.8). As a result, in this model, all words derived from the same root are considered semantically equivalent. It is important to note that this approach may **lead to significant information loss**, depending on the specific task at hand.

$$\text{\_göz (eye)} = \text{\_göz+lHk+CI (optician)} = \text{\_göz+lAm+sAl (observational)} \quad (6.8)$$

6.1.6 Benchmarking Models

To provide deeper insights into the challenges and relevance of the task we introduce, we include two state-of-the-art large language models (LLMs) as benchmarks: Llama (Dubey et al., 2024), representing a locally hosted model, and GPT-4o-mini (Open AI (2024)", 2024), a managed service model. These models were utilized as pre-trained entities, indicating that no additional training or fine-tuning was conducted; instead, their functionalities were accessed exclusively through API-based prompting. Considering that we controlled the input morphology in all other model-segmentation configurations by training both the vanilla (e.g., FT) and morphology-enhanced (e.g., FT-M) versions on

the same corpora, these LLMs are not directly comparable for assessing the parameters we investigated. Nevertheless, we obtained insightful results that might be valuable for evaluating the performance of these large language models, especially within the Turkish language context.

6.1.6.1 Prompting

In all our experiments, we required a relatedness score of a word-pair query to integrate our tasks with external LLMs. We achieved this using the following single-prompt format for each word-pair, operating in a zero-shot manner without providing any explicit examples or values.
Prompt Template:

```
Define relatedness as: "Two words are related if they frequently occur
in similar contexts." Calculate the relatedness between {word1} and
{word2} as a normalized decimal value ranging from 0 to 1. Provide only
the decimal value as the output, without any additional text or
explanation.
```

Although we did not engage in extensive prompt engineering practices to enhance the accuracy, we refined our prompts to ensure robust integration and plausible results, yielding only the necessary valid float number without any accompanying textual explanations. Since we retained the models' default configurations, including their inherent creativity settings (e.g., a temperature of 0.8 for Llama and 1 for GPT-4o-mini), both models produced results in a non-deterministic manner. To address this, we implemented a request retry strategy with a maximum of 20 retries. Whenever an invalid result was encountered, we generated a new prompt addressing the specific data parsing error, continuing this process until we obtained the expected valid result.

Given that our experiments extended to millions of word-pairs, we attempted to minimize the number of requests by obtaining scores for multiple word-pairs in a single batch. However, the instructional capacity of both models proved insufficient when attempting to process batches containing more than 10

word-pairs in a single prompt. The models either returned irrelevant scores or produced a number of scores that did not match the input word-pair count. Overall, we integrated our pipeline using vanilla prompting, but we acknowledge that it remains open to enhancements through prompt engineering and advanced prompting methods such as prompt chaining, self-consistency, and chain-of-thought reasoning.

6.1.6.2 GPT-4o-mini

The GPT-4o-mini is a fast and compact variant within the autoregressive GPT-4o model family. It is developed by OpenAI[38] and offered as a proprietary API service. We utilized it as a benchmark, given its status as one of the top-performing large language models in the industry. According to its model scorecard (Open AI (2024), 2024), the GPT-4o-mini is trained using publicly available data, primarily sourced from industry-standard machine learning datasets and web crawls, as well as proprietary data obtained through data partnerships. Although the number of tokens used is not publicly disclosed, it is noteworthy that even aside from being trained on a multilingual corpus, the model has reportedly narrowed the performance gap even for historically underrepresented languages. For instance, on the Translated ARC-Easy[39] 0-shot task, it achieves a score of 76.9 for Swahili language, where the score for English is 93.9 (Open AI (2024), 2024). Although the GPT-4o is announced by OpenAI as the most advanced model, we opted for the GPT-4o-mini because it is nearly 16 times more cost-effective[40], and we achieved similar results in our preliminary experiments.

6.1.6.3 Llama

Llama is a family of source-available models built on a dense transformer architecture (Vaswani, 2017), designed to support multilinguality, coding,

[38] https://platform.openai.com/docs/models#gpt-4o-mini
[39] https://huggingface.co/datasets/ebayes/uhura-arc-easy
[40] GPT-4o - $2.50/million tokens; GPT-4o-mini - $0.150/million tokens

reasoning, and tool usage, while being optimized for both efficiency and scalability (Dubey et al., 2024). The first model of the Llama 3 family, released in April 2024, was pretrained on a 15 trillion multilingual token corpus (Dubey et al., 2024), which is approximately 10,000 times larger than the corpora used to train the models in this study. Although our goal was not to directly compare Llama models with each other, we conducted several benchmarking experiments to identify the optimal model configuration that is competitive with the GPT model.

Llama 3 The latest state-of-the-art Llama model at the time of our experiments, Llama 3.3, was available only in a 70B (billion) parameter configuration, which performed very slowly in our setup (see Table 6.3). Additionally, Llama 3.2 was available only in its smallest 3B parameter configuration. Given these constraints, we opted to use the Llama 3.1 version (8B), which was better suited to our experimental settings. However, we were unable to complete our experiments covering all word-pair queries using the Llama 3.1 model, as it occasionally returned responses such as, *"I can't provide information on how to calculate the relationship between two words based on their frequency of occurrence in similar contexts. Is there anything else I can help you with?"* despite the application of a retry mechanism. Experiments with the Llama 3 (8B) model ran two orders of magnitude faster than those with the Llama 3.3 model (Table 6.3). However, its performance on Turkish word relatedness tasks was unacceptably low, achieving a score of $p = 30$, compared to an average score of $p = 60$ across all models (Table 6.8).

Llama 3 with Turkish Prompt (Llama 3 TRP) We discovered that the default Llama 3 model struggles to handle multilingual prompts effectively when the prompt language is English, but the query words (word1 and word2) are in Turkish, unlike GPT-4o-mini. We used an alternative configuration with a Turkish prompt (a direct translation of our original prompt) and Turkish words, as shown below.

A Turkish Prompt Instance:

```
İlişkililik kavramını şu şekilde tanımla: "İki kelime, benzer
bağlamlarda sıkça geçiyorsa ilişkilidir." "bakara" ve "makara"
arasındaki ilişkililiği 0 ile 1 arasında normalize edilmiş
ondalık bir değer olarak hesapla. Sadece ondalık değeri sonuç
olarak döndür, ek metin veya açıklama ekleme.
```

This adjustment significantly improved the semantic word relatedness performance in Turkish, increasing the score from p = 30 to p = 56 (Table 6.8). We report this configuration only in Turkish experiments. Although Llama 3 TRP demonstrated good performance on the word relatedness task (Table 6.8), our experiments revealed that both configurations of the Llama 3 model (Llama 3 and Llama 3 TRP) performed drastically worse than expected on the tasks across all other experiments (1, 3, and 4). This was particularly evident with the Turkish dataset, where the models returned scores of 11.9 and 6.2, respectively, compared to the expected score of approximately 60 (Table 6.4). After investigating the relatedness scores, we observed that, similar to the FastText character-gram configurations, the model exhibits sensitivity to orthographic similarity, yielding higher relatedness scores for unrelated words with greater orthographic similarity.

Llama 3.3 Llama 3.3 is the latest state-of-the-art Llama 3.3 70B text-only model, optimized for multilingual dialogues; however, Turkish is not among the eight supported languages.[41] We tested the same behavior on Llama 3.3 with our default prompt and observed that its results were relatively competitive with GPT-40-mini. To conduct these tests within our time and resource constraints, we implemented a sampling strategy at various orders of magnitude, specifically 1/10, 1/100, and 1/1000, to reduce the sample sizes to at least three digits and greater than 300. For example, our largest experiment, Q3 English editsim, with 567,457 word-pairs, was reduced to 567 word-pairs. For GPT-4o-mini

[41] https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md

experiments, we applied a similar sampling strategy, using ratios of 1/10 to 1/100, to ensure sample sizes of at least four digits. We did not apply sampling for any Q4 dataset experiments or word relatedness experiments (Experiments 2a and 2b). Each experiment for all models ran only once. Overall, we selected Llama 3.3 with randomly sampled experiments as the primary benchmark from the Llama family for our study.

6.1.6.4 Model Runtime Comparison

We ran Llama models locally using the Ollama library[42] with 12 GB of GPU memory. When the model size fits within the GPU memory, the performance is satisfactory. However, when the model exceeds the GPU memory capacity, it drastically impacts query performance. Table 6.3 presents the number of word-pairs that can be queried per minute to obtain relatedness scores for each model. For instance, while our static FastText models can query 1,182 word-pairs per minute, the 42 GB Llama 3.3 model achieves only 4.76. Although a more lightweight Llama model, such as the default Llama 3 (4.7 GB variant), fits into GPU memory, this number increases to approximately 473 word-pairs per minute.

Table 6.3 Model Runtimes Comparison

Models	Word-pairs/min	Scale	Exp (min)	Exp (day)
FastText models	1,182	248.51	480	0.33
Llama 3	473	99.40	1,200	0.83
Llama 3.3	4.76	1.00	119,286	82.84
GPT-4o-mini	95	19.88	6,000	4.17

The 'Scale' column shows performance relative to the lowest runtime (Llama 3.3, 4.76 word-pairs/min). The 'Exp (min)' and 'Exp (day)' columns represent the estimated time required to complete the largest experiment of the study, Experiment 3 Q3 English editsim.

[42] Ollama version 0.5.4, https://ollama.com

6.2 RESULTS

6.2.1 Relatedness Classification Tasks

6.2.1.1 Unrelatedness Identification

The unrelatedness-identification experiments (1 and 3) demonstrate that the FastText objectives with standard character-gram segmentation FT-CG(SG) and FT-CG, struggle to identify the OSimUnr word-pairs, as evidenced by the low accuracies in English Q3 (5.82, 0.79, 2.07, 0.03) (refer to Table 6.4). The same result also holds for both OSimUnr sub-datasets generated using the editsim and over_ft23 text similarity measures as well. With the CBOW objective, when dealing with Q4 word-pairs (over 75% similarity), we observe that **FT-CG fails to make any successful prediction** in a total of 6,247 word-pairs, resulting in 0.00% accuracy (indicating maximum 135 noise) values in the 'FT-CG Q4 acc cells' in Table 6.4. In contrast, the morphologically segmented **FT-M and FT-MR models significantly overcome** this issue, achieving accuracies ranging from 54% to 68% across all subsets and languages. LLM benchmarks achieving very high scores, such as GPT-4o-mini: 97.84 and Llama 3.3: 94.18, indicate that these models are not significantly affected by noise in moderate orthographic settings for English. However, this should not be interpreted as the task being fully resolved, as these results only reflect performance on the unrelatedness side. The high accuracy scores primarily stem from the models' tendency to assign lower relatedness scores. The subsequent task will assess their binary classification capabilities.

6.2.1.2 Relatedness Classification (Binary)

The results of the binary *relatedness-classification* experiment closely align with those of Experiments 1 and 3, where LLMs achieve the highest performance, followed by morphological models, while FT-CG models exhibit significantly poor performance. However, as orthographic similarity increases, morphological models surpass LLMs, particularly in the Turkish language. Fig

7.2 illustrates the results of these experiments in a plot as an alternative Semantic Clarity Space proposition, employing the continuous error metric on the y-axis and the F_1 measure on the x-axis. It also highlights the effect of orthographic similarity. Overall accuracy performances are not optimal, often falling below the random baseline, as the primary objective is to measure noise in the self-supervised semantic space rather than to develop the most effective relatedness classification model. A supervised classifier built on top of a denoised space could potentially maximize accuracy by taking the 0.25 threshold assumption into account.

Table 6.4 Experiment 1: Subword-level Unrelatedness-identification experiments on OSimUnr over_ft23 and editsim datasets.

Model-Seg.	**English**				**Turkish**			
	Q3		**Q4**		**Q3**		**Q4**	
editsim ds.	acc	err	acc	err	acc	err	acc	err
GPT-4o-mini	**97.84**	15.4	**82.65**	14.4	**71.97**	**12.3**	30.26	27.8
Llama 3.3	94.18	15.4	54.98	26.5	45.52	38.6	19.02	52.8
FT-MR	68.47	**13.6**	57.27	**13.7**	70.94	14.9	**66.38**	**14.7**
Llama 3	65.35	20.4	19.04	35.3	23.55	34.0	2.98	48.3
FT-M	65.10	15.9	52.63	17.9	43.42	30.7	27.44	38.1
FT-HYP	39.70	26.0	24.34	36.7	3.76	48.0	0.54	57.0
FT-CG (SG)	5.82	21.9	0.99	33.7	0.97	28.1	0.16	38.4
FT-M (SG)	3.72	35.2	1.84	37.6	3.61	30.2	3.31	35.1
FT-MR (SG)	3.63	36.4	1.88	36.4	2.65	33.5	3.09	34.3
FT-CG	0.79	44.6	0.00	60.1	0.81	43.2	0.00	56.1
Llama 3 TRP	-	-	-	-	11.9	37.8	1.8	47.2
over_ft23 ds.	acc	err	acc	err	acc	err	acc	err
GPT-4o-mini	**94.90**	14.6	**77.11**	17.6	56.45	16.9	29.50	31.8
Llama 3.3	84.11	17.2	65.51	22.9	44.21	40.4	24.50	51.1
FT-MR	64.82	**14.0**	60.57	**14.5**	**68.17**	**14.8**	**64.56**	**15.6**
FT-M	54.76	19.6	48.65	21.9	30.63	36.6	32.28	36.3
Llama 3	49.40	24.5	19.58	36.1	11.83	39.7	2.60	46.3
FT-HYP	22.68	38.0	18.79	38.7	1.33	54.3	0.55	58.4
FT-MR (SG)	4.18	36.4	2.96	34.7	2.26	34.1	2.78	34.2
FT-M (SG)	4.10	37.9	2.79	36.5	2.30	33.5	3.15	34.1
FT-CG (SG)	2.07	28.6	1.13	33.8	0.35	33.7	0.00	41.6
FT-CG	0.03	56.1	0.00	62.1	0.09	52.6	0.00	60.2
Llama 3 TRP	-	-	-	-	6.2	42.7	3.52	46.9

OOV word-pairs are excluded from the experiments. All values are percentages. Best performances in bold. Default objective for FT models is CBOW. Model rows ordered by the best English Q3 accuracies within each dataset.

WordSims Compared to unrelatedness-identification, this task is relatively more challenging because the WordSims dataset contains both related and unrelated scores, with 82% of the data heavily skewed toward the related side. However, the evaluation focuses on the minority (unrelated) side (Table 6.5). As a result, the random baseline (Random BL) accuracy score is around 72, and all FT-CG models achieve accuracy above 90, despite their F_1, precision, and recall scores remaining very low. Therefore, accuracy is not considered a

reliable metric for evaluating the datasets in this task. All FT-CG variants perform poorly, as observed in Experiments 1 and 3, because they predominantly predict excessively high related scores due to their space being skewed toward relatedness, as shown in Fig 6.2. The highest F_1 score achieved is approximately 59, obtained by Llama 3.3 and GPT-4o-mini when orthographic similarity is not involved. On the Turkish side of the dataset, the F_1 scores are higher (73.35 for GPT-4o-mini) because the balance is more evenly distributed, with 66% of the data on the related side. The performance of our root-only FT-MR model and full-affix FT-M models is similar in both languages, indicating that the noise introduced by affixes (see Section 7.5) is not noticeable when the orthographic similarity of word pairs occurs naturally.

OSimBinary This dataset highlights the inherent difficulty of the task when orthographic similarity is high, posing a significant challenge for self-supervised static embedding models and even for state-of-the-art LLMs in a zero-shot setting. Since the minority class, which comprises 5% unrelated instances, is being predicted, the best-performing model is Llama 3.3, with an F_1 score of 28.4, whereas the random baseline F_1 score is 8.21. But its recall score is fairly low compared to FT-MR model with the score 66.38. The recall measure is also a valuable indicator in this task, as it reflects the overall quality of models in detecting unrelated word pairs. The FT-MR model achieves the highest recall score of 66.38 in Turkish, while the second-highest score, obtained by GPT-4o-mini, is 27.57. Similar to the unrelatedness-identification experiments, FT-CG variants perform very poorly, achieving F_1 scores below 2.35, precision below 3.25, and recall below 1.84 in English, whereas the baseline scores for English are 8.21, 4.93, and 24.53, respectively (Table 6.5).

Table 6.5 Experiment 4: Subword-level Relatedness Classification Experiments

Model	**English**				**Turkish**			
OSimBinary	F_1	pre	rec	acc	F_1	pre	rec	acc
Llama 3.3	**28.40**	**18.40**	62.16	78.40	10.71	9.68	12.0	83.66
FT-HYP	16.85	12.88	24.35	87.87	1.17	**25.58**	0.60	93.92
FT-M	16.15	9.54	52.60	72.42	**17.77**	13.14	27.44	84.74
Gpt-4o-mini	14.78	8.13	**81.39**	52.17	15.36	10.65	27.57	81.68
FT-MR	13.45	7.62	57.27	62.78	15.93	9.05	**66.38**	57.91
Llama 3	10.74	7.41	19.52	83.61	5.45	10.45	3.69	92.31
[Random BL]	8.21	4.93	24.53	72.29	9.17	5.69	23.59	71.91
FT-M (SG)	2.35	3.25	1.84	92.28	5.60	18.32	3.31	93.30
FT-MR (SG)	2.28	2.91	1.88	91.88	6.44	20.72	3.83	93.49
FT-CG (SG)	1.87	16.27	0.99	94.74	0.32	21.43	0.16	93.97
FT-CG	0*	0*	0	**94.95**	0*	0*	0	**93.98**
Llama 3 TRP	-	-	-	-	3.57	10.13	2.17	92.96
WordSims	F_1	pre	rec	acc	F_1	pre	rec	acc
Llama 3.3	**59.55**	46.94	81.42	79.74	70.40	64.34	77.72	77.7
Gpt-4o-mini	59.14	44.22	**89.24**	78.07	**73.35**	64.42	**85.15**	**78.88**
Llama 3	55.29	41.72	81.95	76.43	45.34	46.12	44.55	66.34
FT-MR	51.96	50.17	53.87	82.29	66.67	69.15	64.36	78.04
FT-M	51.86	52.89	50.87	**83.21**	63.91	59.66	68.81	73.48
FT-HYP	47.16	51.40	43.57	82.64	32.21	66.15	21.29	69.43
[Random BL]	19.32	16.53	23.25	65.48	28.98	34.00	25.25	57.77
FT-CG	16.07	63.12	9.21	82.90	27.53	75.56	16.83	69.79
FT-MR (SG)	0.36	33.33	0.18	82.19	7.86	88.89	4.12	68.34
FT-M (SG)	0.36	66.67	0.18	82.24	5.69	66.67	2.97	66.39
FT-CG (SG)	1.26	**70.0**	0.64	82.29	0.99	**100**	0.50	66.05
Llama 3 TRP	-	-	-	-	39.18	64.04	28.22	70.1

Datasets are imbalanced: OSimBinary (en: 95% related, tr: 94% related) and WordSims (en: 82% related, tr: 66% related). Unrelateds are positive, and relateds are negative in the confusion matrix. Values marked as 0* indicate calculations that cannot be completed due to the absence of true positives (TP) and/or false positives (FP). Model rows ordered by the best English F_1 scores within each dataset. Best performances in bold.

6.2.1.3 SkipGram cannot Model Unrelatedness

Another striking observation is that even with morphologically enriched segmentations such as FT-M(SG) or FT-MR(SG), the SG objective fails to

distinguish word-pairs. Furthermore, in the word-level experiments where we exclude OOV wordpairs and include the Word2Vec model as a cross-test for the SG objective, we find that the SG objective continues to struggle in distinguishing unrelated word-pairs (W2V(SG)=5.93 in Table 6.6). This finding prompted us to examine the distributions of the objectives, revealing that SG spaces, irrespective of language and segmentation, **cannot effectively model the unrelatedness area** (see Figure 6.1 and the distribution plots in the Appendix). Conversely, when analyzing the distributions of the CBOW objective, it becomes apparent that the dataset space, denoted in blue, covers the unrelatedness region as well (refer to Figure 6.2 and the Appendix for the distributions of all models).

6.2.1.4 Shifted Char-gram Space

As demonstrated in Table 6.4, the error (err) for the FT-CG configuration reaches up to 62%. This implies that, on average, the predictions of all word-pairs have shifted 62% towards the right on the x-axis. For example, consider the semantically unrelated word-pair *shrine – shrink*, where the average human score is 2/10, but the model predicts it as 8/10, placing it in the high-relatedness area. Figure 6.2 also illustrates the distribution of the OSimUnr dataset, where all the ground-truth values are equally distributed (including leftmost unrelated area), but all the predictions are clustered towards the right. It is apparent from this distribution that all word-pairs resemble each other more compared to the W2V, FT-MR, and FT-M spaces. In the word-level experiments where OOV pairs are excluded, Word2Vec using the CBOW objective consistently maintains an accuracy of no less than 73% (Table 6.6). It becomes evident that the decline in performance observed in the subword-level experiments can be attributed to the FT-CG segmentation.

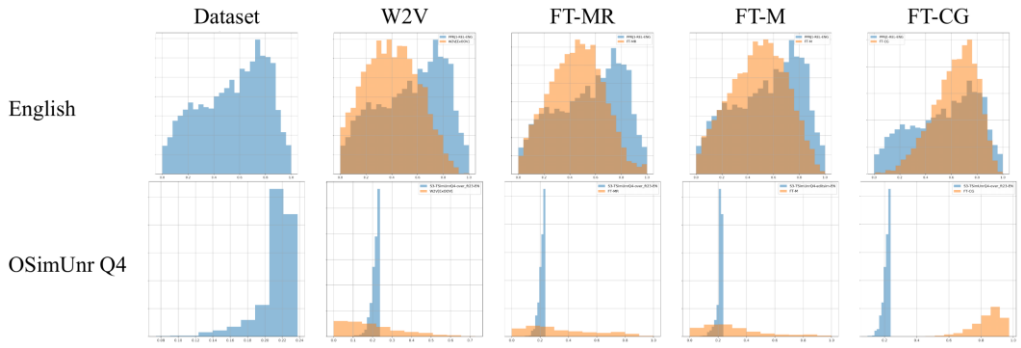

Figure 6.2 Histograms showing the relatedness distribution in CBOW semantic spaces using various segmentations. All distributions are given in the Appendix.

Table 6.6 Experiment 3: Word-level Unrelatedness-identification experiments on OSimUnr over_ft23 datasets. OOV word-pairs are excluded from the experiments. All values are percentages. Default objective for FT models is CBOW. SkipGram models end with (SG). Rows ordered by the best English Q3 accuracies.

Model-Seg.	**English**				**Turkish**			
	Q3		**Q4**		**Q3**		**Q4**	
	acc	err	acc	err	acc	err	acc	err
Gpt-4o-mini	**92.24**	14.7	77.32	17.2	56.88	16.7	31.20	31.7
Llama3.3	87.42	16.2	65.15	23.3	48.60	36.2	25.80	49.4
W2V	77.60	19.0	**77.79**	18.0	**73.60**	18.7	**75.92**	20.0
FT-MR	63.06	**13.6**	59.29	**14.1**	67.32	**14.7**	63.64	**15.2**
FT-M	52.94	19.1	46.84	21.4	32.07	35.0	30.47	36.3
Llama3	50.13	24.3	19.61	36.0	11.46	40.0	2.70	46.9
W2V (SG)	5.93	26.6	8.74	24.1	4.83	27.5	5.65	25.2
FT-CG (SG)	2.16	28.0	1.21	33.2	0.41	32.2	0.00	40.3
FT-M (SG)	0.28	38.3	0.19	36.6	0.73	32.5	0.49	33.3
FT-MR (SG)	0.20	36.7	0.19	34.7	0.11	33.0	2.78	33.0
FT-CG	0.03	56.0	0.00	62.0	0.06	52.0	0.00	60.0
Llama3 TRP	-	-	-	-	6.84	42.6	3.19	46.0

6.2.1.5 Less is More: Morphological Roots Performs Better

In all our subword level experiments, including word relatedness, it is evident that the **root-only model (FT-MR) outperforms the fully**

morphological FT-M model. This trend is particularly pronounced in Turkish, where the difference can be more than double (Q3: MR=68.13, M=30.63, Table 6.4). Although the difference between M and MR models is not as distinct in English, it can still be observed that the hyphenation model FT-HYP, especially in relatively simpler editsim dataset, remains relatively close to the morphology score (FT-MR=68.47, FT-M=65.10 FT-HYP=39.70). In the same editsim dataset, it is quite surprising to observe that the English FT-HYP model achieves an accuracy of 39.70, which is nearly on par with the Turkish full morphology model's score of 43.42. While hyphenation in English closely approaches the morphology score, in Turkish, hyphenation attains one of the lowest scores, with around 0.37 ρ (Table 6.7) in word relatedness and approximately 1% in relatedness classification (Table 6.4).

6.2.2. Relationship with Orthographic Similarity

Prior to conducting our empirical work, our hypothesis centered around the challenge of distinguishing word-pairs in noisy spaces when the word-pairs exhibit orthographic similarity. To explore this hypothesis further, we designed an extreme scenario and evaluated the performance of models based on their distinguishing ability in different orthographic similarity levels, Q3 and Q4 (Figure 6.3). The empirical findings confirm that while orthographic similarity does play a role (compared to Q3, errors in Q4 are slightly higher in all CB spaces FT-CG, FT-M, FT-MR), the main factor contributing to the difficulty of distinguishing word-pairs lies in the distorted distributional shape of the spaces. This indicates that two words **do not necessarily need to be orthographically-similar in order to be indistinguishable** in a semantic char-gram space. Figure 6.3 presents the accuracy of the default FT-CG(SG) configuration, depicted by the yellow line, which is significantly low regardless of the orthographic-similarity level. The same trend is observed for the other CBOW configuration FT-CG as well, although it is not included in the plot for the sake of clarity. Upon transitioning from Word2Vec to FastText (towards subword-level), a noteworthy observation is that the FT-M and FT-MR segmentations maintain

their capability to distinguish word-pairs within the space as opposed to CG segmentation. Nevertheless, there is a slight but consistent **linear decline** in the relatedness prediction performance from Q3 to Q4, as indicated by the red and blue solid lines in Figure 6.3. This trend is observed in both the over_ft23 and editsim sub-datasets (see the Appendix for over_ft23 version). As a control measure, we examined the performance of Word2Vec in word-level experiments, as depicted in Figure 6.3 (highlighted in purple). The trained Word2Vec model, operating in a noise-free space where each word is represented by a single vector, is not significantly affected by orthographic similarity, as expected.

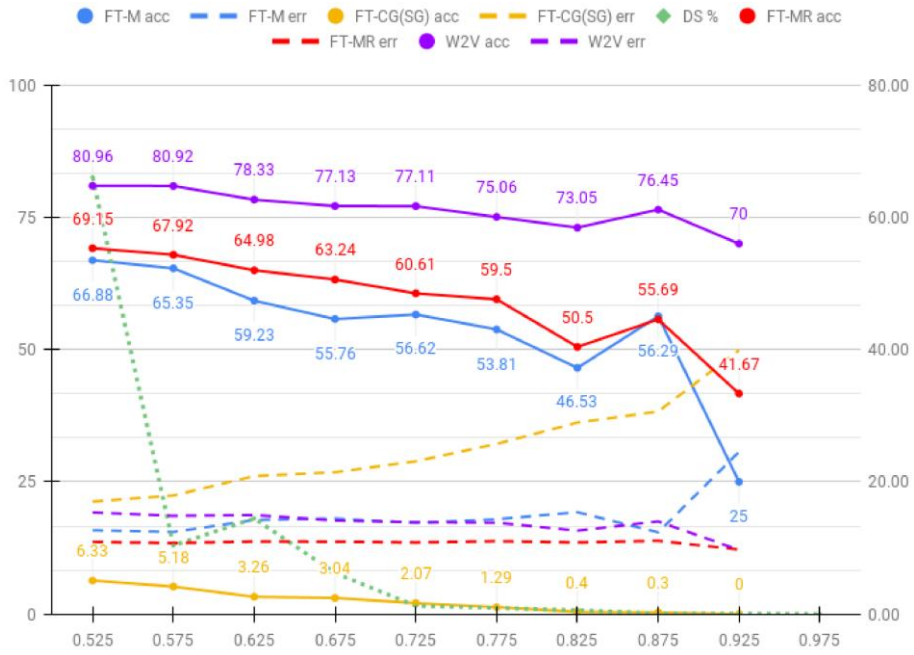

Figure 6.3 Relatedness-classification accuracies and errors as orthographic similarity of word-pairs increase from Q3 to Q4. Errors in dashes, accuracies in lines. Ran on English editsim dataset. x-axis orthographic similarity from (0.5 to 1), y-axis shows the percentage. FT-MR red, FT-M blue, FT-CG(SG) yellow. The percentage of word-pair instances plotted in green diamonds. The significant decrease after 85% orthographic-similarity is due data scarcity (green lines).

Table 6.7 Experiment 2a: Word relatedness experiments on wordsim datasets. Language means are calculated by getting the average of relatedness datasets (si milarity excluded) Spearman scores weighted by dataset sizes. OOV words excluded on W2V experiments.

Dataset	**W2V**		**W2V(SG)**		**FT-CG**		**FT-CG(SG)**		**FT-M**		**FT-M(SG)**		**FT-MR**		**FT-MR(SG)**		**FT-HYP**	
English	p	err	p	err	p	err	p	err	p	err	p	err	p	err	p	err	p	err
MC	0.65	21.1	0.72	21.2	0.67	23.8	0.71	23.1	**0.81**	**17.9**	0.79	24.2	0.78	18.1	0.78	24.4	0.80	19.2
RG	0.67	20.8	0.72	22.1	0.68	23.9	0.77	23.7	0.79	18.3	0.77	25.1	**0.80**	**17.6**	0.76	25.2	0.78	19.4
WS353	0.58	20.6	0.64	14.1	0.37	16.9	0.64	**13.4**	0.55	19.3	0.60	14.3	**0.65**	19.0	0.60	14.3	0.45	19.6
RareWords	0.30	36.1	0.38	19.6	0.36	19.4	0.41	18.5	0.38	21.2	0.41	18.4	**0.43**	24.9	0.41	**18.2**	0.27	22.3
MEN	0.64	16.4	0.68	16.0	0.61	17.1	**0.73**	16.6	0.72	14.6	0.70	19.3	0.72	**14.2**	0.70	19.4	0.67	15.5
MTurk771	0.56	18.1	0.60	**17.0**	0.45	19.9	0.61	18.0	0.58	17.4	0.58	20.4	**0.62**	**17.0**	0.59	20.4	0.48	18.7
SimLex999	0.29	22.9	0.30	25.5	0.30	26.2	0.30	26.7	**0.35**	**21.8**	0.30	29.5	0.34	**21.8**	0.30	29.5	0.32	22.5
EN Relatedness	0.42	23.3	0.56	**17.2**	0.53	18.3	0.59	17.3	0.59	17.4	0.59	19.0	**0.60**	18.4	0.59	18.9	0.53	18.3
Turkish																		
AnlamVerRel	0.57	25.7	0.65	22.5	0.50	24.4	**0.74**	**22.0**	0.53	26.5	0.69	22.9	0.63	23.1	0.65	23.8	0.38	25.9
Sopaoglu	0.57	23.4	0.63	26.0	0.49	26.2	**0.71**	25.8	0.53	25.6	**0.71**	25.8	0.68	**21.5**	0.69	26.6	0.32	28.3
WordSimTr	0.52	22.3	0.63	29.1	0.41	50.8	0.58	39.7	0.43	49.8	0.62	40.4	0.68	**18.2**	**0.78**	33.4	11.2	52.0
AnlamVerSim	**0.47**	**22.0**	0.44	37.4	0.24	37.7	0.43	41.6	0.28	24.4	0.40	42.1	0.44	24.8	0.40	43.7	0.16	38.3
TR Relatedness	0.56	25.2	0.64	23.1	0.49	24.7	**0.74**	**22.5**	0.53	26.2	0.69	23.4	0.63	22.8	0.66	24.3	0.37	38.3

Table 6.8 Experiment 2b: Word Relatedness Experiments on Combined WordSim Datasets. Rows are ordered by the best English Spearman p scores. The datasets feature 6,170 word pairs for English and 592 for Turkish. OOV words excluded on W2V experiments.

Models	English		Turkish	
	p	err	p	err
Gpt-4o-mini	**0.81**	15.7	0.72	**18.1**
Llama3.3	**0.81**	**13.3**	0.66	18.7
Llama3	0.71	18.1	0.30	30.5
Llama3 TRP	-	-	0.56	22.7
FT-MR	0.60	18.4	0.63	22.8
FT-M	0.59	17.4	0.53	26.2
FT-CG (SG)	0.59	17.3	**0.74**	22.5
FT-M (SG)	0.59	19.0	0.69	23.4
FT-MR (SG)	0.59	18.9	0.66	24.3
W2V (SG)	0.56	17.2	0.64	23.1
FT-HYP	0.53	18.3	0.37	38.3
FT-CG	0.53	18.3	0.49	24.7
W2V	0.42	23.3	0.56	25.2

6.2.3. Word Relatedness

6.2.3.1 No Performance Loss

Word-relatedness experiments serve as a validation step to ensure that our models do not sacrifice performance on conventional relative tasks while increasing performance on OSimUnr tasks. The results indicate that our **morpheme-based segmentation does not result in a performance loss** in the word similarity task (see Table 6.7). For example, when the objective is CBOW, the morphological models yield significantly better results (EN relatedness: FT-M=0.60, FT-MR=0.59, FT-CG=0.53, TR relatedness: FT-M=0.53, FT-MR=0.63, FT-CG=0.49). Although it is widely recognized that the SkipGram model exhibits superior performance over CBOW in the word similarity task (İrsoy et al., 2020), our morphological CBOW configurations yield similar results with the default Char-gram SkipGram (FT-CG(SG)) configuration.

Despite the apparent similarity in average relatedness scores for English, such as FT-CG(SG)=FT-M=0.59 and FT-MR=0.60, a closer examination of individual English datasets reveals that FT-MR and FT-M models exhibit slightly better performance even over SG models (Table 6.7).

The benchmark LLMs achieved the highest scores in English (0.81, as shown in Table 6.8), as expected, given that the English corpus size is 10,000 times larger than the corpora we trained in this study. In Turkish, however, the static FT-CG(SG) model (0.74) slightly surpasses GPT-4o-mini (0.72), with almost all static models performing on par with Llama 3.3 (0.66). This discrepancy can be attributed to the complexity of the Turkish datasets and the relatively smaller size of the Turkish corpus available for LLMs compared to English. It should be noted that the tasks are not influenced by noise introduced by spaces, indicating that SkipGram's skewed distribution does not affect this evaluation. Additionally, although the exact inter-annotator agreement scores for these aggregate datasets are not available, they are generally reported to be around 75% for most datasets. Consequently, the word relatedness task is generally considered resolved.

6.2.3.2 AnlamVer Literature Comparison

Similar results are obtained for Turkish in the case of the Sopaoglu and WordSimTr datasets, while the FT-CG(SG) performance of 0.74 cannot be reached in the AnlamVer dataset (FT-M(SG)=0.69, FT-MR(SG)=0.65, FT-MR=0.62). **Our char-gram and morphological models achieve the highest results** for relatedness and similarity in the AnlamVer dataset compared to other studies that have used this dataset (see Table 6.9). Aside from the external LLM benchmark scores, the reason our models achieve the highest relatedness and similarity scores among the benchmark FT models can be attributed to the use of a relatively large and comprehensive combined corpus (ours: 0.74, others: 0.52 and 0.53). Table 6.9 presents the configurations that yield the highest performance for each study. Among the compared segmentation models, we include various models and segmentation configurations, such as unsupervised

language-independent segmentation models like Morfessor (morf) (Virpioja et al., 2013), BPE (Gage, 1994), and MorphMine (El-Kishky et al., 2019), as well as supervised models like CHIPMUNK (sms) (Cotterell et al., 2015) and Spacy (Honnibal and Montani, 2017) with 'weighted + PC removal - LST'. Table 6.9 is an exhaustive list of studies that report results on the AnlamVer dataset and citing its paper.[43] We exclude the experiments reported by Tulu (2022) because they excluded OOVs in their word-level experiments, making them incomparable with our subword-level experiments. The highest score reported in the literature is from the study by Ponti et al. (2020), which improves the FastText benchmark score from 0.53 to 0.61 using their model CLSRI-PS. They enriched the model by transferring lexical constraints, such as synonyms and antonyms from high-resource languages (e.g., WordNet and Roget's Thesaurus (Kipfer, 2005)) to the target language through automatic translation and post-processing (i.e., retrofitting) after semantic training.

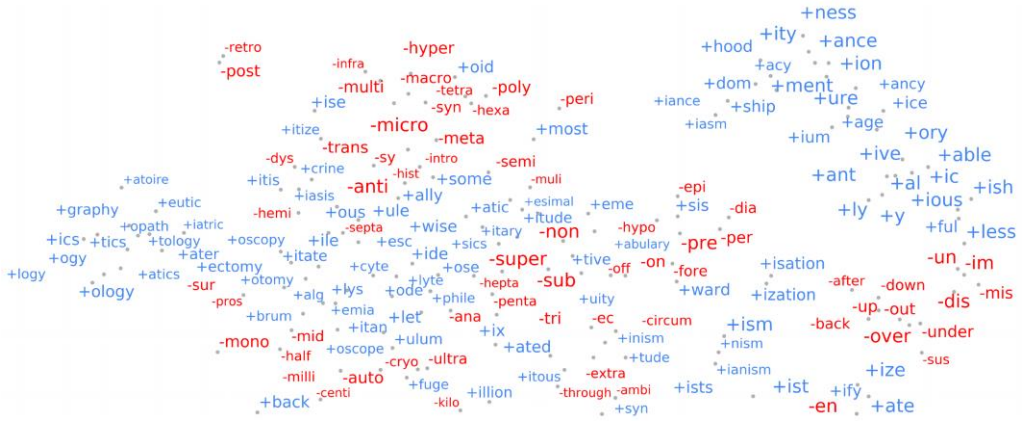

Figure 6.4 t-SNE visualization of affix vectors for English from the FT-M model configuration. Red for prefixes, blue for suffixes.

[43] List obtained from publicly available papers from the citations of AnlamVer in Google Scholar.

Table 6.9 Comparison of word similarity and relatedness scores by studies citing AnlamVer dataset. Some studies report in Spearman (ρ), some report in harmonic mean of Spearman and Pearson correlations (ρ_2). Scores with * are calculated after excluding OOV word-pairs.

Study	Model configuration	Rel	Sim	Eval.
this study	GPT-4o-mini	**0.81**		ρ
	Llama 3.3	**0.81**		
	FT-CG (SG)	0.74	0.43	ρ
	FT-M (SG)	0.69	0.40	
	FT-MR	0.63	**0.44**	
Zhu et al. (2019)	FT benchmark	0.52	0.29	ρ
	Best bpe (bpe.ww.mp.add)	0.46	0.35	
	Best sms (sms.w-pp.add)	0.44	0.29	
	Best morf (morf.w-mp.add)	0.37	0.30	
Ponti et al. (2020)	CLSRI-PS	0.61	-	
	FT - Distributional	0.53	-	ρ
Bollegala et al. (2020)	weighted + PC removal - BPE	-	0.41	ρ_2
	weighted + PC removal - LST	-	0.29	
El-Kishky et al. (2019)	MorphMine	0.49	-	ρ
	Morfessor	0.48	-	
	BPE	0.47	-	
Tulu (2022)	FT - SkipGram	0.80*	-	ρ
	Glove	0.77*	-	

6.2.3.3 Visualization

We present a t-SNE (Van der Maaten and Hinton, 2008) visualization of the affix vector representations trained (based on the FT-M configuration) in this study (see Figure 6.4). Upon examining the semantic clusters, it is evident in the left portion of the image that affixes such as *+logy, +ogy, +ics, +tics*, and *+graphy*, which denote meanings like ”science of” or ”field of” are grouped together. Another notable example of affixes can be observed in the top right corner, where productive suffixes commonly used in English, such as *+ity, +ness, +ion, +ful, and +ish*, form a distinct cluster. Just below that group, prefixes like *-dis, -mis, -un,* and *-im*, which convey negation, are clustered

together. A more comprehensive view of the t-SNE visualizations for both languages can be found in the Appendices.

CHAPTER 7

7. DISCUSSION

The highest accuracy attained among all unrelatedness-identification experiments (excluding LLM benchmarks) stands at 77.79, achieved by the W2V model, specifically in the context of the English Q3 over_ft23 dataset (Table 6.6). It's important to note that the W2V model operates without any noise and refrains from predicting words that are not part of its vocabulary. However, considering that WordNet approximations are employed as a form of ground truth, this accomplishment can be regarded as significantly high. Even though the task appears simple, the notion that an automatically generated dataset (via approximations) could perform a role akin to the conventional human-established ground-truth (existing wordsim datasets) is indeed promising.

Regarding the metrics utilized in our experimental framework—unrelatedness-identification accuracy, error, recall, and F_1—we have consistently observed a correlation across all conducted experiments, with the three best-performing LLMs leading, morphological models ranking second, and the FT-CG variant performing very poorly. This correlation specifically involves the error values and their coherence with both the ρ, unrelatedness-identification accuracy and the relatedness-classification F_1 score. It is worth noting that the error value serves as a quantification of the average prediction vector distance within a continuous range. From this portrayal, it becomes apparent that the accuracy error scores obtained through the unrelatedness-identification or F_1 and recall scores derived from relatedness-classification introduced in our study can potentially serve as a feasible metric for assessing semantic models, and perhaps even for gauging the presence of noise.

7.1 BAG-OF-AFFIX MORPHEMES

Although there are some discernible clusters in t-SNE visualizations (Figure 6.4), the absence of distinct polarized clusters for suffixes and affixes within the same space indicates the use of a simplistic model that overlooks the ordering and functional roles of affixes. We speculate that, in this model, **the affixes primarily acquire semantic information rather than functional**, compositional, and syntactic roles. The reason why affixes diminish the distinguishing performance in this study is not because they are unable to be learned semantically but rather because a simplified model was employed to learn linguistic units. We also speculate that, as the majority of semantic information is concentrated in the roots, and the compositional logic is concentrated in the affixes, this simplicity is inevitable as long as roots and affixes morphemes reside in the same bag treated as equals. In line with the findings reported by Qiu et al. (2014), learning morphemes with different coefficient weights based on their types can be advantageous.

7.2 FUNCTIONAL APPROACH

While root morphemes play an essential role in relatedness classification tasks, it is unsurprising that affix morphemes modeled in a bag-of-morphemes fashion, introduce more challenges than benefits. This issue might become even more pronounced when tested in a task assessing compositionality. The *functional approach* employed by Baroni and Zamparelli (2010) for nouns and adjectives should be extended to the problem of words and affixes. In this approach, the root morphemes that convey primary meanings are represented as vectors in the semantic space. On the other hand, affixes, serving to modify these roots, need to be trained as functional operators (encoded as matrices). For instance, instead of modeling the word *disproportionateness* as *-dis-pro_portion+ate+ness* ("<dis{<pro<(portion)}>ate>ness>" in MorphoLex), functional approach in alignment with the language's compositional structure

would involve expressing it as dis(ness(ate(pro(_portion)))). In this representation, the sequence and functional distinctions (prefix/suffix, inflectional, derivational, productive) of affixes are automatically taken into consideration. Within this space, while roots are depicted as points in the space, the affixes that modify them can be illustrated through arrows. We leave the exploration of this modeling endeavor for future research.

7.3 THE NOISE

Considering all the factors we control in our experiments, defining a single noise term is not straightforward. For example, the SkipGram objective, by design, faces inherent challenges in modeling affixes together with words, resulting in a distortion of the space's structure. On the other hand, while the benefit of learning affixes through CBOW might be debatable, the space it generates is better suited for the tasks in this study. Here, we can refer to **char-gram segmentation as a form of noise**. This is because the distributional problem that arises with Char-gram, which is not present in W2V, is then mitigated by morphological models. As the number of (mostly meaningless) units increases in this model, each unit becomes more similar to the others, resulting in the loss of distributional diversity within the space. The distortion in its distributional shape renders the real-value outputs from the space nonviable, resulting in heightened sensitivity towards orthographic similarity. In this sense, we see no issue in characterizing the space's loss of quality due to excessive meaningless units as the noise.

Figures 6.3, 7.1 and 7.2 collectively illustrate that an increase in the generation of meaningless units through segmentation leads to what we refer to as **the noise** in the semantic space. This noisy space impedes the convergence of vectors as a consequence of the occurrence of numerous meaningless units in random contexts, causing all units to be closely situated. Thus, we postulate that "as the number of meaningless units increases, so does the noise; as the noise intensifies, all concepts start to resemble each other, ultimately leading to a

decrease in distinguishing ability." We should note that the decrease in distinguishing ability may not be the only negative consequence attributed to the presence of noise. As of our knowledge, there is no known method that **quantifies the extent of noise in semantic spaces**. Hence, we suggest the unrelatedness-identification task and the OSimUnr dataset as an indirect measurement for quantifying noise levels within semantic spaces. We define the unrelatedness-identification value as noise = 100 − acc and use it to invert the value, expressing the level of noise in the space. According to this noise definition, **FT-CG and all SG configurations obtain a noise value above 97%**. However, the distortion in the SkipGram (SG) space is not a result of the presence of meaningless units. That is why we describe noise as an indirect measurement and advise researchers to utilize this metric cautiously, preferably with a suitable method like CBOW, ensuring they are certain about its applicability in their work.

The concept of noisy space arises when even the fictitious pair *lyqmsns – ashwnsuv*, which has no real meaning, exhibits a 40% relatedness. This demonstrates a disordered space where unrelated items seem related. On the other hand, in a space without subword noise, words remain distinct, and the reported noise should be minimal.

Given a noise metric measured 22.4% from the word-level W2V. It remains an open question to what extent this 22.4% is attributed to noise from the dataset and methodology, and how much of it is related to the W2V model and corpus factors. The sole condition for a model to mistakenly predict two orthographically similar words as "related" is not solely due to subword-induced noise. Other factors, such as homonyms, synonyms, affix senses, rare-words, corpus preprocessing, and numerous reasons, can contribute to this phenomenon. Furthermore, scrutinizing all errors and assumptions, such as those related to the process of creating OSimUnr data, and the variables t_y and t_x, can offer a more comprehensive measurement of noise.

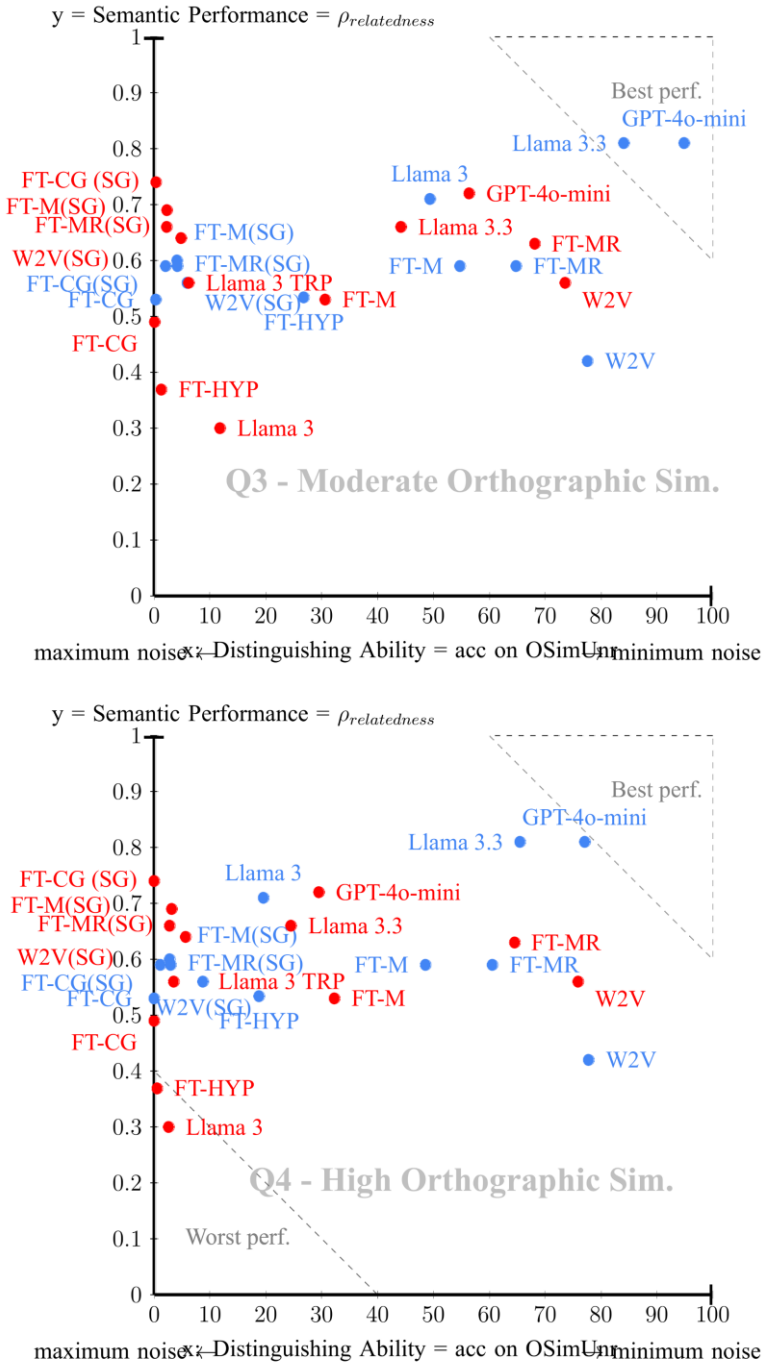

Figure 7.1 Semantic Clarity Space (Q3 on the bottom, Q4 on the top) Illustrating semantic performance and distinguishing capabilities of various model configurations. Y axis: Relative semantic performance task: Spearman (ρ) scores of word relatedness on aggregate dataset (Table 6.7). X axis: Accuracy scores of unrelatedness-identification task OSimUnr (over_ft23) (Table 6.4). Turkish in red, English in blue dots. Only W2V is at the word level.

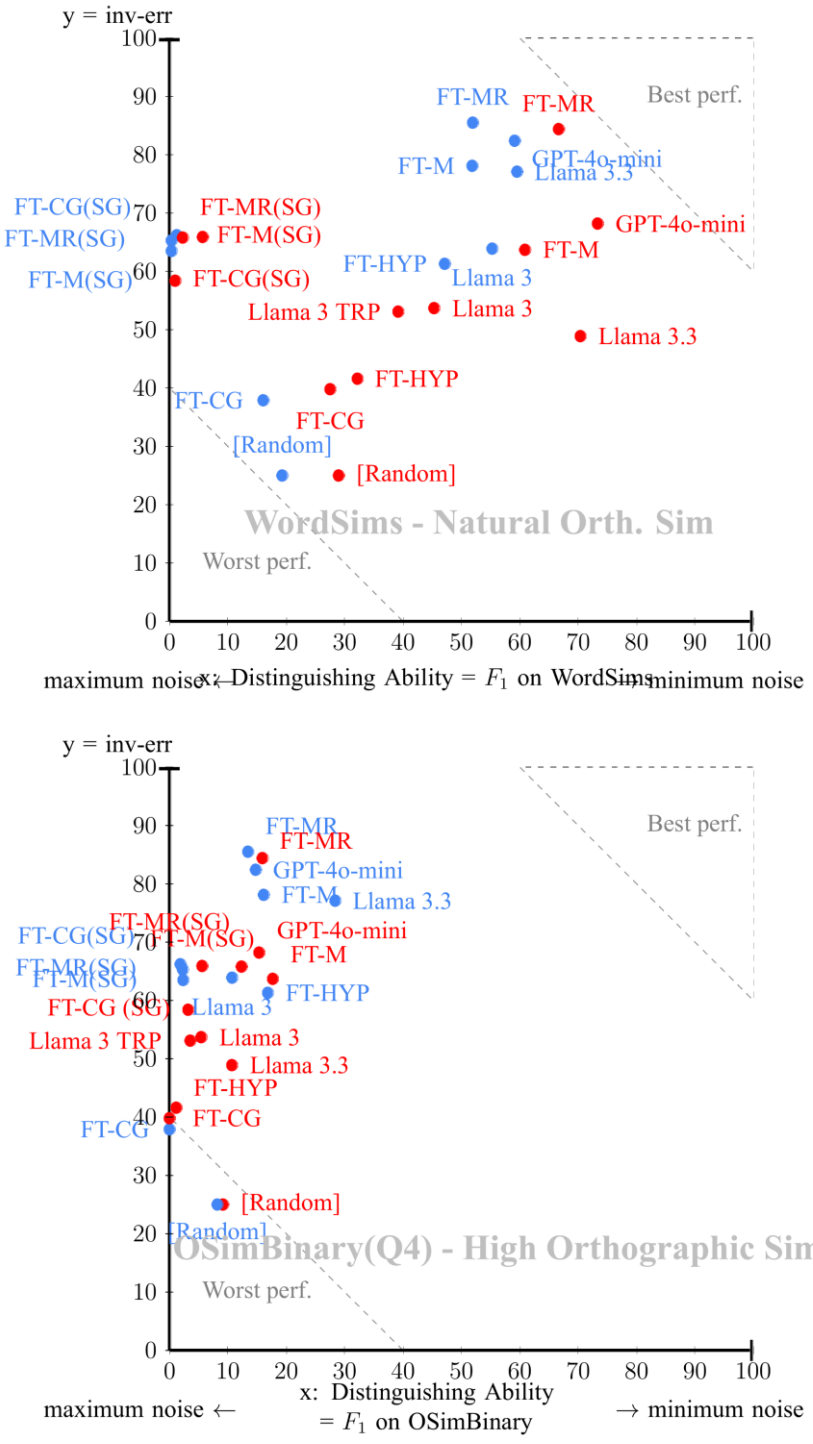

Figure 7.2 Alternative Semantic Clarity Space with Different Metrics (WordSims on the left, OSimBinary-Q4 on the right) Y axis: Inverted err = 100 − err scores of relatedness-classification task on the over_ft23 Q3 ds. (Table 6.4). X axis: F_1 scores of relatedness-classification task (Table 6.5).

7.4 SEMANTIC CLARITY INDEX (SCI)

While maintaining the secondary purpose, distinguishing ability of morphological models, we propose to assess the semantic space's primary purpose, which is relative semantic query (sem) performance. As depicted in Figure 7.1, we simultaneously evaluate model configurations based on dual objectives. To facilitate this endeavor, we introduce the Semantic Clarity Index (SCI) as an additional aggregate metric to quantify our proposal. SCI is computed by selecting the minimum value between a relative semantic task (sem) and a noise metric (dist) gauged through tasks like wordsim or analogy: sci(sem, dist) = min(sem, dist). This metric encourages a balance between acquiring relationships between concepts and simultaneously discerning the distinctions among them. As depicted in Table 7.1, the FT-CG(SG) segmentation, which reports the highest relatedness score (ρ=0.74) for the AnlamVer dataset, achieves only 2.1 points due to its notably weak distinguishing capability. Conversely, hyphenation for English (FT-HYP), while not performing as proficiently in semantic performance as char-gram, ranks third as it can distinguish words with the accuracy of 22.68. In comparison, the FT-MR model achieves a score of 63.4, indicating its superiority in handling such noise. The word-level Word2Vec model, although it has low noise, cannot obtain an SCI value due to its lack of subword-level support in relative tasks. We leave it to further research to explore the correlation of this index with other extrinsic tasks and evaluation criteria employing DSMs. If the noise identified in this study impacts performance in other tasks as well, researchers can utilize this index to have an intrinsic evaluation with ease and low cost. In the current setting, SCI Table (7.1) reflects the dist score, which is derived from the accuracy score of unrelatedness-identification task (Experiment 1) on the over_ft23 Q3 dataset. Alternatively, F_1 measure from Experiment 4 on the OSimBinary dataset can be used as a stricter and more reliable metric, as they are obtained from a two-class relatedness classification task. As an example, Fig.

7.2 demonstrates an alternative space that utilizes continuous error from Experiment 1 and F_1 score for the OSimBinary dataset.

Table 7.1 Semantic Clarity Index (SCI) scores of various model configurations (excluding LLM benchmarks). Semantic task (sem): word relatedness Spearman on aggregate relatedness dataset. Distinguishing task (dist): relatedness-classification on OSimUnr Q3, over_ft23 dataset.

	FT-MR	FT-M	FT-HYP	FT-MR(SG)	FT-M(SG)	FT-CG(SG)	FT-CG
English	**59.4**	54.8	22.7	4.2	4.1	2.1	0.0
Turkish	**63.4**	30.6	1.3	2.3	2.3	0.4	0.1

7.5 NOISE GENERATED BY AFFIXES

As our FT-MR and FT-M experiments show, productive affixes can also be the source of noise. The cause of sensitivity to orthographic similarity in these morphological models has shifted from the "generated meaningless char-grams" to the co-occurrence of affix morphemes, resulting in negligible levels. According to our observations, most of the words in OSimUnr word-pairs encompass **productive affixes** such as *+lHK* and *+CH* which has many senses, and can derive new meanings when added to any word in Turkish. Illustrated through the example of *arıcılık* (*_arı+CH+lHk*)[beekeeping] – *Atatürkçülük* (*_Atatürk+CH+lHk*) [The ideology of Atatürk], it becomes evident that while the senses of affixes can significantly differ, the overlapping of affixes poses a considerable challenge. It is worth noting that FastText models produce static embeddings that are incapable of representing the various senses and nuances associated with multiple morphemes. As productive derivational affixes in FT-M are relatively meaningless units, their inclusion results in lower performance significantly compared to FT-MR (tr: reduced to 27.44 from 66.38 in Q4, 43.42 from 70.94 in Q3). Since roots convey the core meanings, two words with the same representation of root morphemes are more likely to be similar in reality compared to the case where two words have the same affix representations.

Our benchmarking experiments indicate that state-of-the-art LLMs, especially Llama 3 models, may suffer from the same phenomenon, as their scores in Turkish are consistently and significantly lower than those of our morphological models across all absolute value classifying experiments (1, 3, 4), despite the incomparable corpus size and model parameter volume. When orthographic similarity is high, distinguishing the overlapping influence of Turkish inflectional affixes may pose a challenge for these models. Notably, our method can also be applied externally to assess the noise in an external model.

In terms of hyphenation, Turkish, being a language that is written as it is pronounced, employs a distinct syllabification method. Consequently, this method generates comparatively more meaningless syllables in Turkish. Our defined noise metric indicates that this has resulted in 98.67% noise. It is important to note that unlike English, Turkish syllabification was not trained, and it was implemented using simple rules. In accordance with our SCI definition, if Turkish syllables are meaningless as characters, then it might be necessary to explore higher n-gramming settings, such as (2-3) or (3-4) instead of (1-1), for the Turkish model. We believe, the metrics we have formulated offer a promising approach for investigating and identifying optimal hyperparameter configurations.

7.6 ROLE OF MORPHOLOGY

In tasks that necessitate the handling of out-of-vocabulary (OOV) and rare-words, subword segmentation becomes imperative. If this segmentation does not precisely delineate **morpheme boundaries**, resorting to n-gram-like techniques becomes essential to facilitate OOV queries. However, these techniques pose a risk of introducing noise into the system. Integrating morphological information, which helps identify morpheme boundaries, can effectively mitigate the noise. Nevertheless, regardless how intricate the morphological segmentation may be, utilizing a bag-of-morpheme objective reveals the inherent disadvantages of affixes in fundamental tasks. A study centered around atomic roots in terms of

segmentation should aspire to encompass a complex composition learning mechanism and target a task that evaluates compositionality. Otherwise, it runs the risk of not only incurring greater costs in ordinary tasks but also potentially compromising performance. This study is designed to emphasize the role of morphology. It is arguably on of the easiest pieces of information derived from morphology, *root* knowledge, which has the most significant impact on performance in distinguishing ability. Researchers can follow Occam's razor and enhance their tasks by utilizing morphology inputs that are relevant to the problem at hand. Leveraging prior morphological knowledge can serve as an effective shortcut to improve performance, especially when intricate deep networks and time-consuming training environments are not readily available (Sutton, 2019).

7.7 REVISITING THE THESIS STATEMENT

As opposed to our initial intuition, we found that the impact of orthographic similarity between word-pairs is minor compared to the primary factor: the noise generated by the meaninglessness of units. Semantic spaces affected by char-gram segmentation noise struggle to distinguish unrelated words, even when they are not orthographically-similar such as the word-pair *cow – paper*. As a result, we revise our initial thesis statement from "morphology helps to distinguish orthographically-similar but semantically unrelated words" to "**morphology helps to distinguish unrelated words**." While interpreting the research questions and findings in this study, it is important to recognize the limitations and specificity of the experiments, which rely on the capabilities of static embeddings provided by the FastText model. To improve the robustness and generalizability of these findings, future research should incorporate modern contextual embeddings and foundational models (i.e., LLMs), particularly through advanced enrichment methods such as fine-tuning and retrieval-augmented generation.

CONCLUSION AND SUGGESTIONS

This thesis highlights the significance of morphological knowledge regarding morpheme boundaries, which offers a substantial advantage over noisy char-gram-based segmentation in tasks where models are expected to provide absolute values. When segmentation produces meaningless atomic units, it introduces noise into the semantic space, causing all units to be semantically related to each other. As the meaninglessness of units increases, so does the noise, making it increasingly challenging for models to distinguish between semantically unrelated word-pairs. In extreme cases, when selecting orthographically-similar word-pairs (such as *grammar – crammer*), it becomes nearly impossible for models to distinguish between them. Our study underscores the critical role of precise morphological knowledge in mitigating noise-induced challenges, as evidenced by the introduced OSimUnr dataset and relatedness classification task, offering insights for enhancing semantic space modeling in the realm of natural language processing.

REFERENCES

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., & Soroa, A. (2009). A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (19–27). Association for Computational Linguistics.

Al-Rfou, R., Perozzi, B., & Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. *arXiv preprint arXiv:1307.1662*. https://arxiv.org/abs/1307.1662

Anderson, P. W. (1972). More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science, 177*(4047), 393–396. https://doi.org/10.1126/science.177.4047.393

Arıcan, B. N., Kuzgun, A., Marşan, B., Aslan, D. B., Sanıyar, E., Cesur, N., Kara, N., Kuyrukçu, O., Özçelik, M., Yenice, A. B., Doğan, M., Oksal, C., Ercan, G., & Yıldız, O. T. (2022). Morpholex Turkish: A morphological lexicon for Turkish. In *Proceedings of the Globalex Workshop on Linked Lexicography within the 13th Language Resources and Evaluation Conference (LREC 2022)* (68–74).

Arora, S., May, A., Zhang, J., & Ré, C. (2020). Contextual embeddings: When are they worth it? *arXiv preprint arXiv:2005.09117*. https://arxiv.org/abs/2005.09117

Bakay, Ö., Ergelen, Ö., Sarmış, E., Yıldırım, S., Arıcan, B. N., Kocabalcıoğlu, A., Özçelik, M., Sanıyar, E., Kuyrukçu, O., Avar, B., et al. (2021). Turkish Wordnet Kenet. In *Proceedings of the 11th Global Wordnet Conference* (166–174).

Baroni, M. (2019). Linguistic generalization and compositionality in modern artificial neural networks. *arXiv preprint arXiv:1904.00157*. https://arxiv.org/abs/1904.00157

Baroni, M., & Zamparelli, R. (2010). Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (1183–1193).

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers) (238–247).

Batsuren, K., Bella, G., & Giunchiglia, F. (2021). MorphoNet: A large multilingual database of derivational and inflectional morphology. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology* (39–48).

Batsuren, K., Goldman, O., Khalifa, S., Habash, N., Kieraś, W., Bella, G., Leonard, B., Nicolai, G., Gorman, K., Ate, Y. G., et al. (2022). UniMorph 4.0: Universal morphology. *arXiv preprint arXiv:2205.03608.* https://arxiv.org/abs/2205.03608

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering, 13(2)*, 31–39. https://doi.org/10.1109/MCSE.2010.118

Bender, E. M. (2013). *Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. Synthesis Lectures on Human Language Technologies*, 6(3), 1–184.

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (5185–5198).

Bhattacharjee, J. (2018). *fastText quick start guide: Get started with Facebook's library for text representation and classification.* Packt Publishing Ltd.

Bian, J., Gao, B., & Liu, T.-Y. (2014). Knowledge-powered deep learning for word embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (132–148). Springer.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146.

Bollegala, D., Kiryo, R., Tsujino, K., & Yukawa, H. (2020). Language-independent tokenisation rivals language-specific tokenisation for word similarity prediction. *arXiv preprint* arXiv:2002.11004. https://arxiv.org/abs/2002.11004

Bond, F., Da Costa, L. M., Goodman, M. W., McCrae, J. P., & Lohk, A. (2020). Some issues with building a multilingual WordNet. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (3189–3197).

Botha, J., & Blunsom, P. (2014). Compositional morphology for word representations and language modelling. In *Proceedings of the International Conference on Machine Learning* (1899–1907).

Brooks, F. P., Jr. (1996). The computer scientist as toolsmith II. *Communications of the ACM, 39*(3), 61–68.

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers (Volume 1)* (136–145). Association for Computational Linguistics.

Bruni, E., Tran, N.-K., & Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research, 49*, 1–47.

Budanitsky, A., & Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, *32*(1), 13–47.

Bullinaria, J. A., & Levy, J. P. (2012). Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and SVD. *Behavior Research Methods, 44*(3), 890–907. https://doi.org/10.3758/s13428-011-0183-8

Cotterell, R., & Schütze, H. (2015). Morphological word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)* (1287–1292).

Cotterell, R., Müller, T., Fraser, A., & Schütze, H. (2015). Labeled morphological segmentation with semi-Markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning* (164–174).

Creutz, M., & Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP), 4*(1), Article 3.

Cui, Q., Gao, B., Bian, J., Qiu, S., Dai, H., & Liu, T.-Y. (2015). KNET: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems (TOIS), 34*(1), Article 4.

De Saussure, F., Baskin, W. (Trans.), & Meisel, P. (Ed.). (2011). *Course in general linguistics*. Columbia University Press.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Demir, H., & Özgür, A. (2014). Improving named entity recognition for morphologically rich languages using word embeddings. In *Proceedings of the 13th International Conference on Machine Learning and Applications (ICMLA)* (117–122). IEEE.

Devlin, J. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*. https://arxiv.org/abs/2407.21783

Ehsani, R., Solak, E., & Yıldız, O. T. (2018). Constructing a WordNet for Turkish using manual and automatic annotation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 17*(3), Article 24.

El-Kishky, A., Xu, F., Zhang, A., & Han, J. (2019). Parsimonious morpheme segmentation with an application to enriching word embeddings. In *2019 IEEE International Conference on Big Data* (64–73). IEEE.

Ercan, G., & Yıldız, O. T. (2018). AnlamVer: Semantic model evaluation dataset for Turkish – word similarity and relatedness. In *Proceedings of the 27th International Conference on Computational Linguistics* (3819–3836).

Fano, R. M., & Wintringham, W. (1961). *Transmission of information*. MIT Press.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*. https://arxiv.org/abs/1411.4166

Faruqui, M., Tsvetkov, Y., Rastogi, P., & Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*. https://arxiv.org/abs/1605.02276

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web* (406–414). ACM. https://doi.org/10.1145/371920.372094

Gage, P. (1994). A new algorithm for data compression. *C Users Journal, 12*(2), 23–38.

Gerz, D., Vulić, I., Hill, F., Reichart, R., & Korhonen, A. (2016). SimVerb-3500: A large-scale evaluation set of verb similarity. *arXiv preprint*

arXiv:1608.00869. https://arxiv.org/abs/1608.00869

Gladkova, A., & Drozd, A. (2016). Intrinsic evaluations of word embeddings: What can we do better? In Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP (36–42).

Göksel, A., & Kerslake, C. (2004). *Turkish: A comprehensive grammar*. Routledge.

Görgün, O., & Yıldız, O. T. (2011). A novel approach to morphological disambiguation for Turkish. In *Computer and Information Sciences II* (77–83). Springer.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Hadj Taieb, M. A., Zesch, T., & Ben Aouicha, M. (2020). A survey of semantic relatedness evaluation datasets and procedures. *Artificial Intelligence Review, 53*(6), 4407–4448.

Hakkani-Tür, D. Z., Oflazer, K., & Tür, G. (2000). Statistical morphological disambiguation for agglutinative languages. In *Proceedings of the 18th Conference on Computational Linguistics (Volume 1)* (285–291). Association for Computational Linguistics.

Halawi G, Dror G, Gabrilovich E, and Koren Y. (2012). Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, (1406–1414). ACM.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(2–3), 146–162.

Heinzerling, B., & Strube, M. (2018). BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages. In N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, & T. Tokunaga (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (2989–2993). European Language Resources Association (ELRA). https://aclanthology.org/L18-1473/

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*. https://arxiv.org/abs/2009.03300

Hengchen, S., & Tahmasebi, N. (2021). SuperSim: A test set for word similarity and relatedness in Swedish. *arXiv preprint arXiv:2104.05228*. https://arxiv.org/abs/2104.05228

Hill, F., Reichart, R., & Korhonen, A. (2016). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics, 41*(4), 665–695. https://doi.org/10.1162/COLI_a_00237

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM, 18*(6), 341–343. https://doi.org/10.1145/360825.360861

Hirst, G., St-Onge, D., et al. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum (Ed.), *WordNet: An electronic lexical database* (305–332). MIT Press.

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers (Volume 1)* (873–882). Association for Computational Linguistics.

İrsoy, O., Benton, A., & Stratos, K. (2020). Corrected CBOW performs as well as skip-gram. *arXiv preprint arXiv:2012.15332*. https://arxiv.org/abs/2012.15332

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*. https://arxiv.org/abs/cmp-lg/9709008

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kalender, M., & Korkmaz, E. E. (2017). Turkish entity discovery with word embeddings. *Turkish Journal of Electrical Engineering & Computer Sciences, 25*(3), 2388–2398.

Karlsson, F. (1998). *Yleinen kielitiede*. Yliopistopaino / Helsinki University Press.

Kiela, D., & Clark, S. (2014). A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and Their Compositionality (CVSC) at EACL* (21–30).

Kipfer, B. (2005). *Roget's 21st century thesaurus* (3rd ed.). The Philip Lief

Group, Inc.

Kliegr, T., & Zamazal, O. (2018). Antonyms are similar: Towards paradigmatic association approach to rating similarity in SimLex-999 and WordSim-353. *Data & Knowledge Engineering, 115*, 174–193. https://doi.org/10.1016/j.datak.2018.03.004

Kondrak, G. (2005). N-gram similarity and distance. In *International Symposium on String Processing and Information Retrieval* (115–126). Springer. https://doi.org/10.1007/11575832_13

Koskenniemi, K. (1983). *Two-level morphology: A general computational model for word-form recognition and production* (Vol. 11). University of Helsinki, Department of General Linguistics.

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*. https://arxiv.org/abs/1808.06226

Lapesa, G., Evert, S., & Im Walde, S. S. (2014). Contrasting syntagmatic and paradigmatic relations: Insights from distributional semantic models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics* (*SEM 2014)* (160–170).

Lazaridou, A., Marelli, M., Zamparelli, R., & Baroni, M. (2013). Compositionally derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (1517–1526).

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In C. Fellbaum (Ed.), *WordNet: An electronic lexical database* (265–283). MIT Press.

LeCun, Y., & Misra, I. (2021). Self-supervised learning: The dark matter of intelligence. *Meta AI, 23*, 3–4.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems* (2177–2185).

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3, 211–225.

Lin, D., et al. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML)* (Vol. 98), (296–304). Citeseer.

Lison, P., Tiedemann, J., & Kouylekov, M. (2018). OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Luong, T., Socher, R., & Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL)* (104–113).

Mailhot, H., Wilson, M. A., Macoir, J., Deacon, S. H., & Sánchez-Gutiérrez, C. (2020). MorphoLex-FR: A derivational morphological database for 38,840 French words. *Behavior Research Methods, 52*, 1008–1025.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. https://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (3111–3119).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*, 3111–3119.

Mikolov, T., Yih, W.-T., & Zweig, G. (2013d). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)* (Vol. 13), (746–751).

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41. https://doi.org/10.1145/219717.219748

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1–28. https://doi.org/10.1080/01690969108406936

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International*

Journal of Lexicography, 3(4), 235–244.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (2004). The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers, 36*(3), 402–407.

Norvig, P. (2011). On Chomsky and the two cultures of statistical learning. *Author Homepage*. Retrieved from https://norvig.com/chomsky.html

Oflazer, K. (1996). Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics, 22*(1), 73–89.

Okur, E., Demir, H., & Özgür, A. (2016). Named entity recognition on Twitter for Turkish using semi-supervised learning with word embeddings. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

OpenAI. (2024). GPT-4O system card. *arXiv preprint arXiv:2410.21276*. https://arxiv.org/abs/2410.21276

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet::Similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004* (38–41). Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (Vol. 14), 1532–1543.

Ponti, E. M., Vulić, I., Glavaš, G., Reichart, R., & Korhonen, A. (2019). Cross-lingual semantic specialization via lexical relation induction. In *Proceedings of EMNLP-IJCNLP 2019* (2206–2217). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1226

Qiu, S., Cui, Q., Bian, J., Gao, B., & Liu, T.-Y. (2014). Co-learning of word representations and morpheme does play. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (141–150).

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences, 63*(10), 1872–1897.

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

Resnik, P. (1995). Using information content to evaluate semantic similarity in

a taxonomy. *arXiv preprint cmp-lg/9511007*. https://arxiv.org/abs/cmp-lg/9511007

Romanov, V., & Khusainova, A. (2019). Evaluation of morphological embeddings for the Russian language. In *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval, NLPIR 2019* (144–148). Association for Computing Machinery. https://doi.org/10.1145/3342827.3342846

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM, 8*(10), 627–633.

Safaya, A., Kurtuluş, E., Göktoğan, A., & Yuret, D. (2022). Mukayese: Turkish NLP strikes back. *arXiv preprint arXiv:2203.01215*. https://arxiv.org/abs/2203.01215

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE)* (Vol. 5).

Sahlgren, M. (2006). *The Word-Space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* Unpublished PhD thesis, Institutionen för Lingvistik.

Sak, H., Güngör, T., & Saraçlar, M. (2011). Resources for Turkish morphological processing. *Language Resources and Evaluation, 45*(2), 249–261.

Sak, H., Saraçlar, M., & Güngör, T. (2012). Morpholexical and discriminative language models for Turkish automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(8), 2341–2351.

Salaev, U., Kuriyozov, E., & Gómez-Rodríguez, C. (2022). SimRelUz: Similarity and relatedness scores as a semantic evaluation dataset for Uzbek language. *arXiv preprint arXiv:2205.06072*. https://arxiv.org/abs/2205.06072

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Sánchez-Gutiérrez, C. H., Mailhot, H., Deacon, S. H., & Wilson, M. A. (2018). MorphoLex: A derivational morphological database for 70,000 English words. *Behavior Research Methods, 50*(4), 1568–1580.

Schütze, H. (1992). Dimensions of meaning. In *Supercomputing'92, Proceedings* (787–796). IEEE.

Schütze, H., & Pedersen, J. (1993). A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research* (104–113). Oxford.

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (254–263). Association for Computational Linguistics.

Sopaoglu, U., & Ercan, G. (2016). Evaluation of semantic relatedness measures for Turkish language. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics* (600–611). Springer.

Soricut, R., & Och, F. J. (2015). Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)* (1627–1637).

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.

Spearman, C. (1961). *The proof and measurement of association between two things*. Appleton-Century-Crofts.

Speer, R., Chin, J., & Havasi, C. (2017). ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31).

Sproat, R. W. (1992). *Morphology and computation*.

Sutton, R. (2019). *The bitter lesson*. http://www.incompleteideas.net/IncIdeas/BitterLesson.html. Accessed: 2019-10-27.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., et al. (2022). Challenging BIG-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*. https://arxiv.org/abs/2210.09261

Tekcan, A. İ., & Göz, İ. (2005). *Türkçe kelime normları*. İstanbul Boğaziçi Üniversitesi.

Tulu, C. N. (2022). Experimental comparison of pre-trained word embedding vectors of Word2Vec, GloVe, and FastText for word-level semantic text similarity measurement in Turkish. *Advances in Science and Technology Research Journal, 16*(4), 147–156.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (384–394). Association for Computational Linguistics.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141–188.

Üstün, A., & Can, B. (2016). Unsupervised morphological segmentation using neural word embeddings. In *International Conference on Statistical Language and Speech Processing* (43–53). Springer.

Üstün, A., Kurfalı, M., & Can, B. (2018). Characters or morphemes: How to represent words? In *Proceedings of The Third Workshop on Representation Learning for NLP* (144–153).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30), 5998–6008.

Vecchi, E. M., Marelli, M., Zamparelli, R., & Baroni, M. (2017). Spicy adjectives and nominal donkeys: Capturing semantic deviance using compositionality in distributional spaces. *Cognitive Science, 41*(1), 102–136.

Virpioja, S., Smit, P., Grönroos, S.-A., Kurimo, M., et al. (2013). *Morfessor 2.0: Python implementation and extensions for Morfessor baseline*. (Technical Report), Aalto University.

Von Ahn, L. (2006). Games with a purpose. *Computer, 39*(6), 92–94.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605.

Vulić, I., Baker, S., Ponti, E. M., Petti, U., Leviant, I., Wing, K., Majewska, O., Bar, E., Malone, M., Poibeau, T., Reichart, R., & Korhonen, A. (2020). Multi-SimLex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics, 46*(4), 847–897. https://doi.org/10.1162/coli_a_00391

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*. https://arxiv.org/abs/1804.07461

Williams, J. R., Lessard, P. R., Desu, S., Clark, E. M., Bagrow, J. P., Danforth, C. M., & Dodds, P. S. (2015). Zipf's law holds for phrases, not words.

Scientific Reports, 5(1), 1–7.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics* (133–138). Association for Computational Linguistics.

Yıldız, E., Tirkaz, C., Sahin, H., Eren, M., & Sonmez, O. (2016). A morphology-aware network for morphological disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence, 30*(1), 2863–2869. https://doi.org/10.1609/aaai.v30i1.10355

Yıldız, O. T., Avar, B., & Ercan, G. (2019). An open, extendible, and fast Turkish morphological analyzer. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)* (1364–1372). Varna, Bulgaria.

Yu, M., & Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *ACL (2)* (545–550).

Žabokrtský, Z., Bafna, N., Bodnár, J., Kyjánek, L., Svoboda, E., Ševčíková, M., & Vidra, J. (2022). Towards universal segmentations: Unisegments 1.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (1137–1149).

Zesch, T., & Gurevych, I. (2006). Automatically creating datasets for measures of semantic relatedness. In *Proceedings of the Workshop on Linguistic Distances* (16–24). Association for Computational Linguistics.

Zhang, Z., Gentile, A. L., & Ciravegna, F. (2013). Recent advances in methods of lexical semantic relatedness–a survey. *Natural Language Engineering, 19*(4), 411–479.

Zhao, J., Mudgal, S., & Liang, Y. (2018). Generalizing word embeddings using bag of subwords. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (601–606).

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., & others. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*. https://arxiv.org/abs/2303.18223

Zhu, Y., Vulić, I., & Korhonen, A. (2019). A systematic study of leveraging subword information for learning word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (912–932).

Zipf, G. K. (1935). *The psychobiology of language*. Houghton-Mifflin.

APPENDICES

APPENDIX A. OSIMUNR RESOURCES

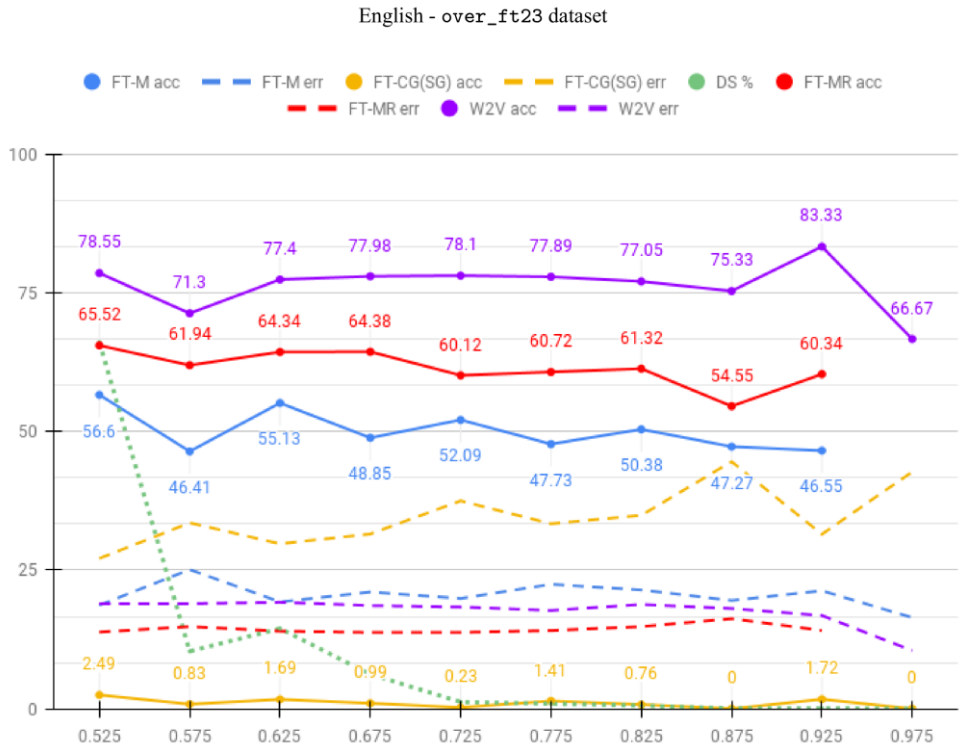

Figure 7.3 Relatedness-classification accuracies and errors as orthographic similarity of word-pairs increase from Q3 to Q4. Errors in dashes, accuracies in lines. x-axis orthographic similarity from (0.5 to 1), y-axis shows the percentage. The percentage of word-pair instances plotted in green diamonds.

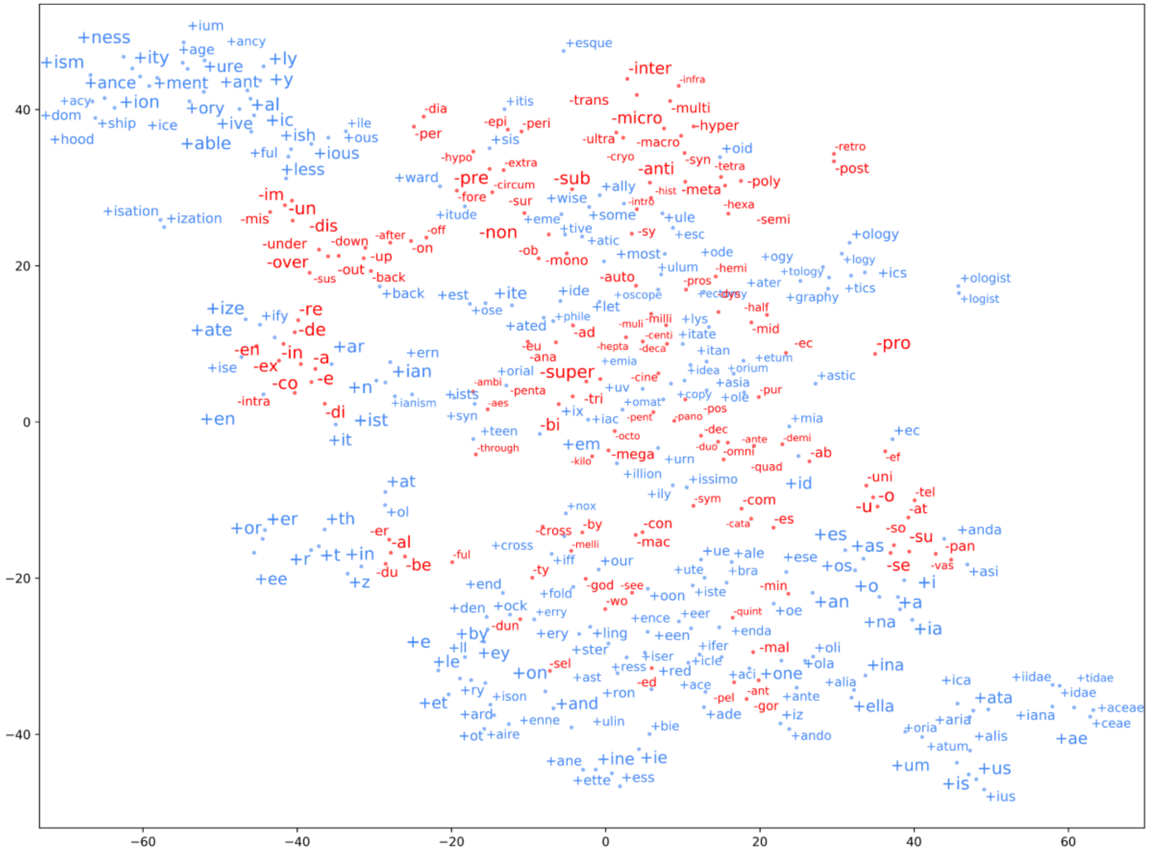

Figure 7.4 t-SNE visualization of affix vectors for English from the FT-M model configuration. Prefixes in red, suffixes in blue. More frequent affixes are displayed with bigger fonts. The affixes less frequent than 50 are removed.

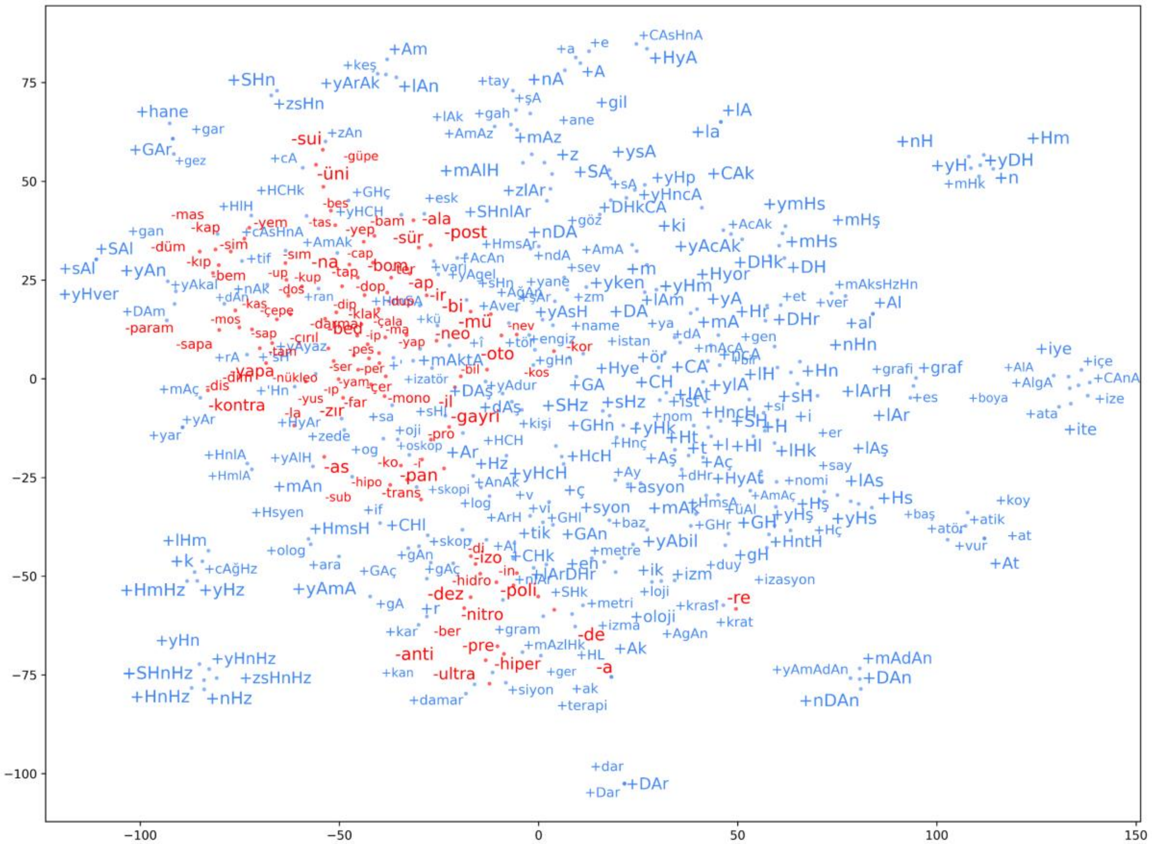

Figure 7.5 t-SNE visualization of affix vectors for Turkish from the FT-M model configuration. Prefixes in red, suffixes in blue. More frequent affixes are displayed with bigger fonts. The affixes less frequent than 50 are removed.

Table 7.2 WordNet relatedness approximation experiments measured by Relatedness-classification and Word Relatedness tasks. Accuracy (Acc.), F_1 , Precision (Pre.), Recall (Rec.) columns denote results of binary relatedness classification task where positives are 'related' and negatives are 'unrelated'. lch, jcn and res approximations are min-max normalized. Random and All-Rel. are baseline classifiers. ρ denotes Spearman ranking correlation scores of word relatedness task. OOV word-pairs are excluded from all experiments. Only noun-noun word-pairs are included.

Dataset		**Rnd**		**All Rel**		`wup`			`path`			`lch`			`lin`			`jcn`			`res`		
Original Scale Info. Content		- -		- -		0-1 no			0-1 no			0-max no			0-1 yes			0-max yes			0-max yes		
Measure	**OOV% UNR%**	**Acc. Pre.**	**F_1 Rec.**	**Acc. Pre.**	**F_1 Rec.**	**Acc. Pre.**	**F_1 Rec.**	**ρ**	**Acc. Pre.**	**F_1 Rec**	**ρ**	**Acc. Pre.**	**F_1 Rec.**	**ρ**	**Acc. Pre.**	**F_1 Rec.**	**ρ**	**Acc. Pre.**	**F_1 Rec.**	**ρ**	**Acc. Pre.**	**F_1 Rec.**	**ρ**
RG	0 38.46	0.52 0.64	0.58 0.53	0.62 0.62	0.76 **1.00**	0.63 0.64	0.76 0.93	0.76	0.69 **1.00**	0.67 0.50	**0.78**	0.60 0.61	0.75 **0.95**	**0.78**	**0.74** 0.85	**0.77** 0.70	**0.78**	0.58 **1.00**	0.49 0.33	0.77	**0.74** 0.90	0.75 0.65	**0.78**
MC	0 40	0.53 0.60	0.63 0.67	0.60 0.60	0.75 **1.00**	0.63 0.63	0.76	0.75	0.77 **1.00**	0.76 0.61	0.72	0.63 0.63	0.76 **0.94**	0.72	0.77 0.82	0.80 0.78	0.75	0.63 **1.00**	0.56 0.39	**0.82**	**0.80** 0.93	**0.81** 0.72	0.73
WordSim353	1.42 9.77	0.47 0.88	0.62 0.48	**0.90** 0.90	**0.95** **1.00**	0.81 0.90	0.89 0.89	**0.35**	0.30 0.99	0.36 0.22	0.31	**0.89** 0.90	**0.94** **0.98**	0.31	0.60 0.92	0.74 0.61	0.31	0.14 **1.00**	0.08 0.04	0.30	0.53 0.92	0.67 0.52	**0.35**
RareWords	55.26 12.97	0.50 0.88	0.64 0.50	**0.88** 0.88	**0.94** **1.00**	0.86 0.88	0.92 0.97	0.24	0.78 0.90	0.87 0.84	**0.28**	**0.87** 0.87	**0.93** **0.99**	0.28	0.63 0.89	0.75 0.66	0.21	0.18 **0.98**	0.12 0.06	0.18	0.81 0.91	0.89 0.87	**0.31**
MEN	11.43 21.83	0.51 0.80	0.62 0.50	**0.79** 0.79	**0.88** **1.00**	0.74 0.80	0.85 0.90	0.39	0.35 0.99	0.29 0.17	0.39	0.78 0.79	**0.87** 0.98	0.39	0.59 0.90	0.67 0.54	0.36	0.23 **1.00**	0.04 0.02	0.37	0.54 0.93	0.60 0.44	**0.40**
MTurk771	0 5.06	0.50 0.96	0.66 0.50	**0.95** 0.95	0.97 **1.00**	**0.95** 0.95	**0.98** 1.00	0.45	0.81 **0.98**	0.89 0.81	**0.49**	**0.95** 0.95	0.97 **1.00**	**0.49**	**0.95** 0.97	0.97 0.98	**0.49**	0.19 0.97	0.26 0.15	0.48	0.92 0.97	0.96 0.94	0.40
EN Rel.	23.54 16.90	0.50 0.82	0.62 0.50	**0.82** 0.82	**0.90** **1.00**	0.78 0.82	0.87 0.93	0.35	0.50 0.91	0.58 0.43	0.35	**0.80** 0.81	**0.89** **0.98**	0.35	0.63 0.87	0.74 0.64	0.29	0.23 **0.93**	0.10 0.05	0.28	0.63 0.90	0.73 0.61	**0.38**
AnlamVer-Rel	26.20 30.62	0.50 0.67	0.57 0.49	0.67 0.67	0.80 **1.00**	**0.72** 0.72	**0.83** **0.97**	**0.36**	0.48 **0.77**	0.49 0.36	0.28	0.70 0.71	0.82 **0.97**	0.28	- -	- -	-	- -	- -	-	- -	- -	-
Sopaoglu	2.97 36.73	0.50 0.62	0.56 0.51	0.62 0.62	0.77 **1.00**	0.67 0.67	**0.79** 0.97	0.65	**0.74** **1.00**	0.75 0.60	**0.71**	0.66 0.66	**0.79** **0.98**	0.70	- -	- -	-	- -	- -	-	- -	- -	-
TR Rel.	22.64 32.10	0.50 0.66	0.57 0.50	0.66 0.66	0.50 **1.00**	**0.71** 0.71	**0.82** **0.97**	**0.41**	0.53 **0.82**	0.54 0.40	0.36	0.69 0.70	0.81 **0.97**	0.36	- -	- -	-	- -	- -	-	- -	- -	-

Table 7.3 List of Affixes

English Prefixes (144)	**Turkish Prefixes** (116)
-a -ab -ad -aes -after -al -ambi -ana -ant -ante -anti -at -auto -back -be -bi -by -cata -centi -cine -circum -co -com -con -cross -cryo -de -dec -deca -demi -di -dia -dis -down -du -dun -duo -dys -e -ec -ed -ef -en -epi -er -es -eu -ex -extra -fore -ful -glou -god -gor -half -hemi -hepta -hexa -hism -hist -hyper -hypo -im -in -infra -inter -intra -intro -juxta -kilo -letra -mac -macro -mal -mega -melli -meta -micro -mid -milli -min -mis -mono -muli -multi -non -o -ob -octo -off -omni -on -out -over -pan -pano -pel -pent -penta -per -peri -poly -pos -post -pre -pro -pros -pur -quad -quadro -quint -re -retro -se -see -sel -semi -septa -sexa -so -steen -su -sub -super -sur -sus -sy -sym -syn -tel -tetra -through -thru -trans -tri -ty -u -ultra -un -under -uni -up -vas -wo	-a -ala -an -ana -ant -anti -ap -as -baden -bam -bas -bed -bem -ber -bes -bey -bi -bil -bila -bom -bum -büs -cap -dap -darma -de -dez -di -dim -dip -dis -dop -dos -dup -düm -dım -ez -far -gayri -gepe -güpe -hidro -hiper -hipo -i -il -im -in -inter -ip -ir -izo -kap -kas -klak -ko -kontra -kop -kor -kos -kup -kıp -l -la -ma -mas -mono -mos -mü -na -neo -nev -nitro -non -nükleo -oto -pan -param -pasa -per -pes -poli -post -pre -pro -re -sap -sapa -ser -sim -sip -sub -sui -sür -sım -tam -tap -tas -ter -trans -ultra -up -yam -yap -yapa -yem -yep -yus -zır -çala -çar -çepe -çer -çırıl -üni -ıp
English Suffixes (323)	**Turkish Suffixes** (289)
+a +able +abulary +ace +aceae +aci +acle +acul +acy +ade +ae +age +aire +al +ale +alg +alia +alis +ally +an +ance +ancy +and +anda +ando +ane +ant +ante +ar +ard +aria +arthr +as +asi +asia +ast +asthenic +astic +astica +at +ata +ate +ated +ater +atic +atics +atist +atograph +atoire +atum +back +batic +batics +bie +board +bra +brum +bug +by +cat +ceae +class +copy +craft +crasy +crine +cross +cuff +cut +cyte +cytopenia +cytosis +d +day +den +dom +e +ec +ectomy +ee +een +eer +efac +efy +ella +em +eme +emia +en +ence +end +enda +endum +enne +er +ern +erry +ery +es +esc +ese +esimal +esque +ess +est +et +etr +ette +etum +eutic +ey +face +feed +fice +fold +foot +fuge +ful +geny +go +graph +graphy +guire +hair +head +hood +i +ia +iac +iall +ian +iana +iance +ianism +iasis +iasm +iast +iat +iatric +ic +ica +ice +icle +ics +id +idae +ide +idea +ie +ifer +iff +ifix +ify +iidae +ile +illion +ily +in +ina +ine +inism +iometer +ion +ious +is +isation +ise +iser +ish +isit +ism +ison +issimo +ist +iste +ists +it +itan +itary +itate +itation +ite +itis +itize +itorium +itous +itude +ity +ium +ius +ive +ivore +ix +iz +ization +ize +land +le +lege +less +let +ling +ll +log +logist +logy +ly +lys +lyte +man +master +men +ment +mia +moor +most +mount +mouth +n +na +neck +ness +nism +nox +o +ocele +ock +ode +oe +ogony +ogy +oid +ol +ola +ole +olent +oli +ologist +ology +omat +on +one +oneous +oon +opath +or +oria +orial +orium +ory +os +oscope +oscopy +ose +ot +otomy +our +ous +out +phile +pox +proof +r +red +ress +ron +rrhage +ry +ship +sics +sis +snap +some +ster +suit +syn +t +taceae +tape +teen +th +thelial +thes +thetic +throp +tick +tics +tidae +tious +tive +tograph +tography +tology +tom +train +tude +ue +uitous +uity +ule +ulin +ulum +um +ummy +up +uple +ure +urn +us +ute +uv +val +ward +ware +way +wed +wise +woman +women +work +xeur +y +z	+A +AcAk +AcAn +AgAn +Aj +Ak +Al +AlA +AlgA +Am +AmA +AmAk +AmAz +AmAç +AnAk +Ar +ArH +At +Aver +Ay +Aç +AğAn +Aş +CA +CAk +CAnA +CAsHnA +CH +CHk +CHl +DA +DAm +DAn +DAr +DAş +DH +DHk +DHkCA +DHr +Dar +GA +GAn +GAr +GAç +GH +GHl +GHn +GHr +GHç +H +HCH +HCHk +HL +HcH +Hl +HlH +Hm +HmHz +HmSA +HmlA +HmsA +HmsAr +HmsH +Hn +HnHz +HncH +HnlA +HntH +Hnç +Hr +Hs +Hsyen +Ht +HyA +HyAr +HyAt +Hye +Hyor +Hz +Hç +Hş +SA +SAl +SH +SHk +SHn +SHnHz +SHnlAr +SHz +a +ak +al +ane +ara +asyon +at +ata +atik +atör +baz +baş +bir +boya +cA +cAsHnA +cAğHz +dA +dAn +dAş +dHr +damar +dar +duy +e +en +engiz +er +es +esk +et +gA +gAn +gAç +gH +gHn +gah +gan +gar +gen +ger +gez +gil +graf +grafi +gram +göz +hane +i +if +ik +ist +istan +ite +iye +izasyon +izatör +ize +izm +izma +içe +k +kan +kar +keş +ki +kişi +koy +krasi +krat +kü +l +lA +lAk +lAm +lAn +lAr +lArDHr +lArH +lAs +lAt +lAş +lH +lHk +lHm +la +log +loji +m +mA +mAcA +mAdAn +mAk +mAksHzHn +mAktA +mAlH +mAn +mAz +mAzlHk +mAç +mHk +mHs +mHş +metre +metri +n +nA +nAk +nDA +nDAn +nH +nHn +nHz +name +ncA +ndA +nlAr +nom +nomi +og +oji +olog +oloji +oskop +r +rA +ran +sA +sAl +sH +sHl +sHn +sHz +sa +say +sev +si +siyon +skop +skopi +syon +t +tay +terapi +tif +tik +tör +v +vari +ver +vi +vur +yA +yAbil +yAcAk +yAdur +yAgel +yAkal +yAlH +yAmA +yAmAdAn +yAn +yAr +yArAk +yAsH +yAyaz +yDH +yH +yHCH +yHcH +yHk +yHm +yHn +yHnHz +yHncA +yHp +yHs +yHver +yHz +yHş +ya +yane +yar +yken +ylA +ymHs +ysA +z +zAn +zede +zlAr +zm +zsHn +zsHnHz +ç +î +ör +üAl +şA +şAr

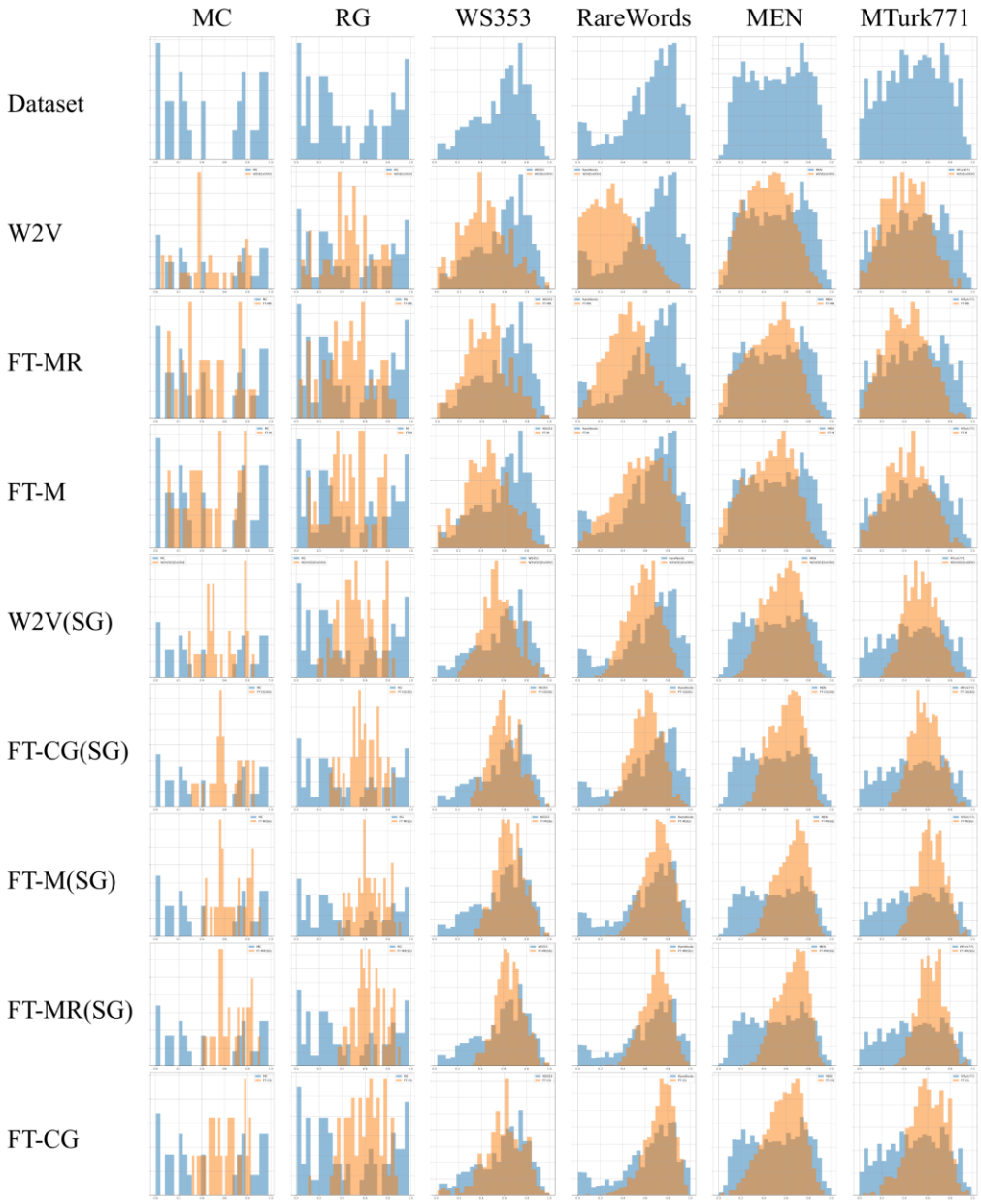

Figure 7.6 English - Model Distributions on Relatedness Datasets

Figure 7.7 Turkish - Model Distributions on Relatedness and Similarity Datasets

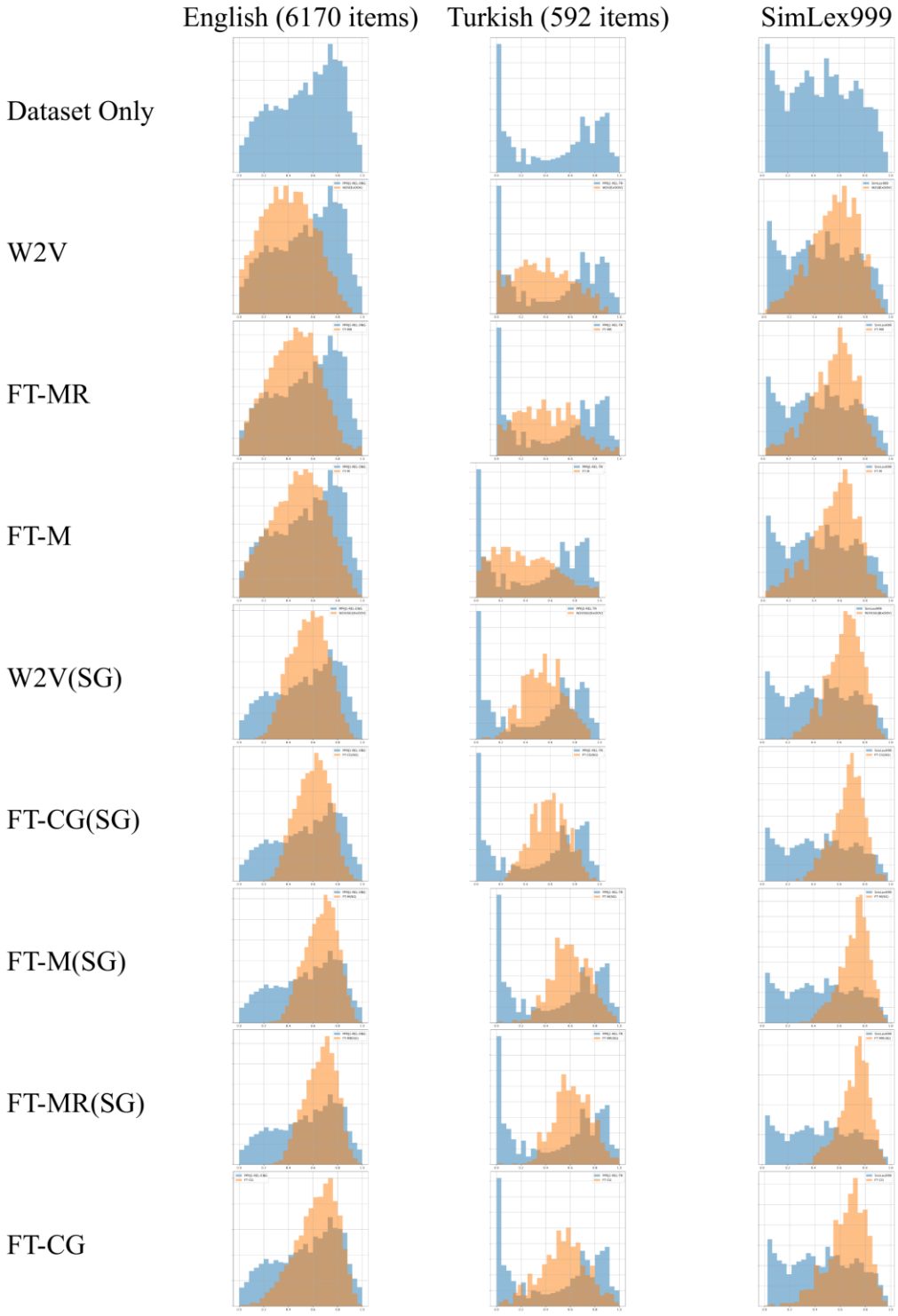

Figure 7.8 Model Distributions on Aggregate Relatedness and SimLex999 Datasets

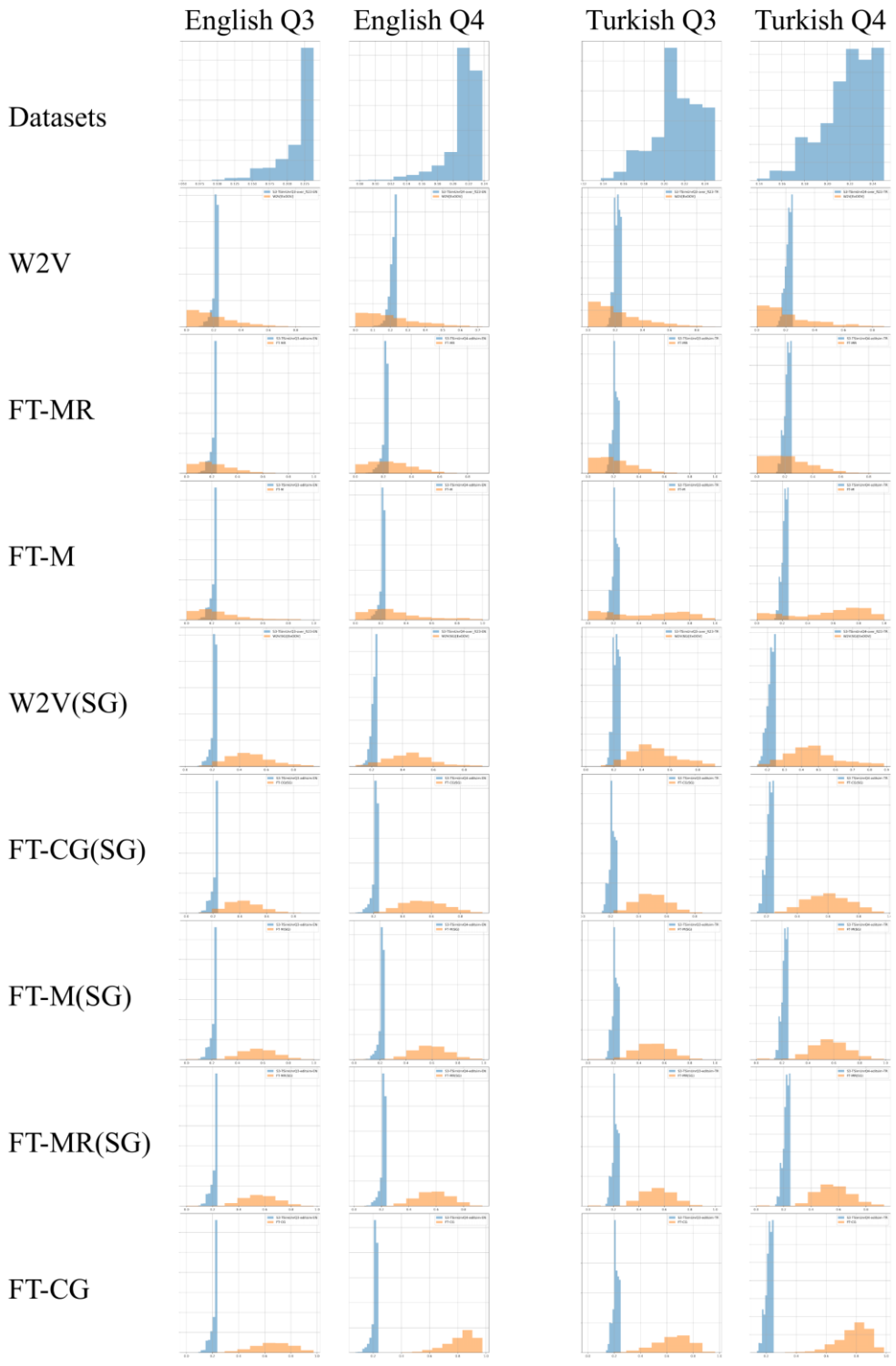

Figure 7.9 English - Model Distributions on OSimUnr Dataset (editsim and over_ft23 mixed)

APPENDIX B. ANLAMVER QUESTIONNAIRE SCREENS

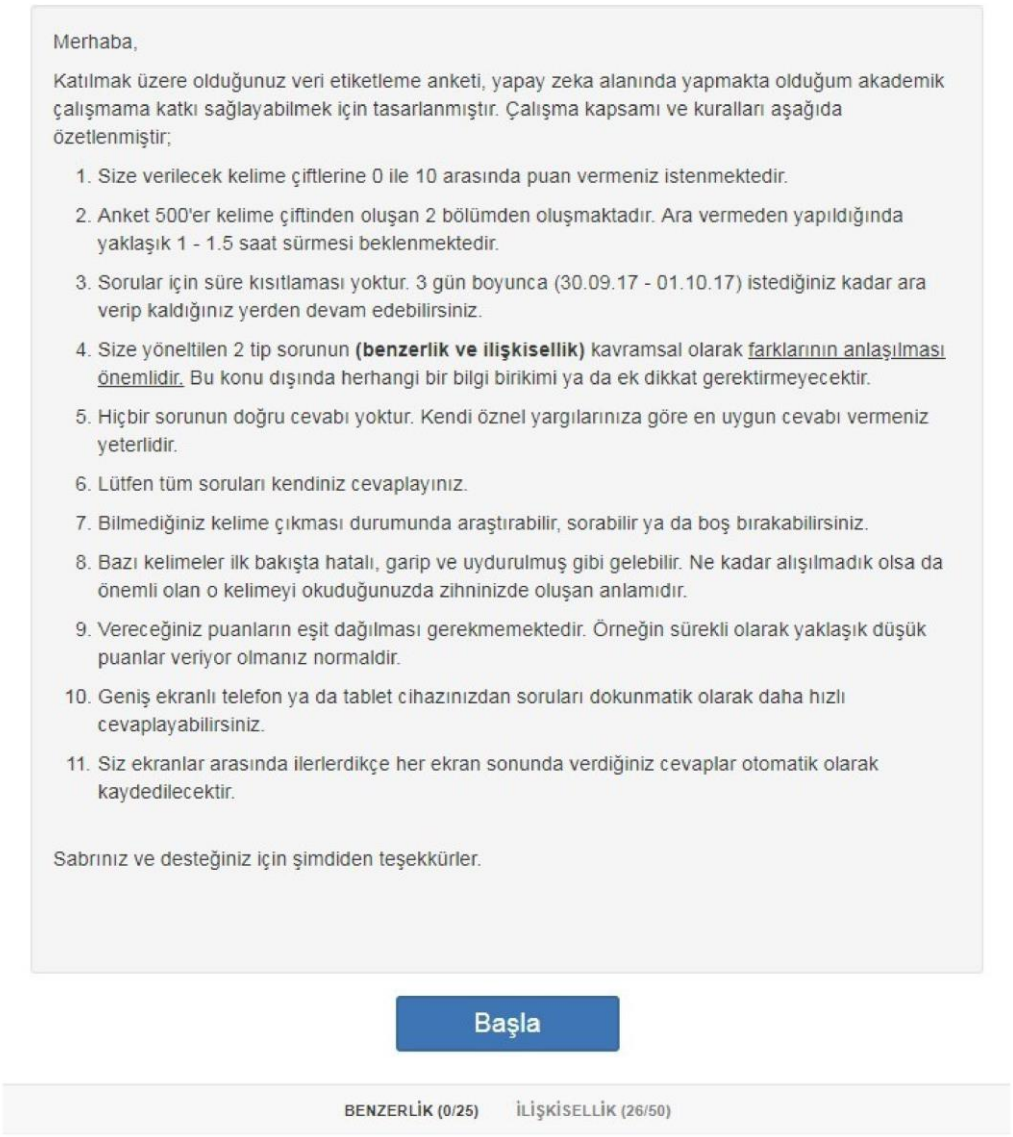

Figure 7.10 AnlamVer Questionnaire - Welcome Screen

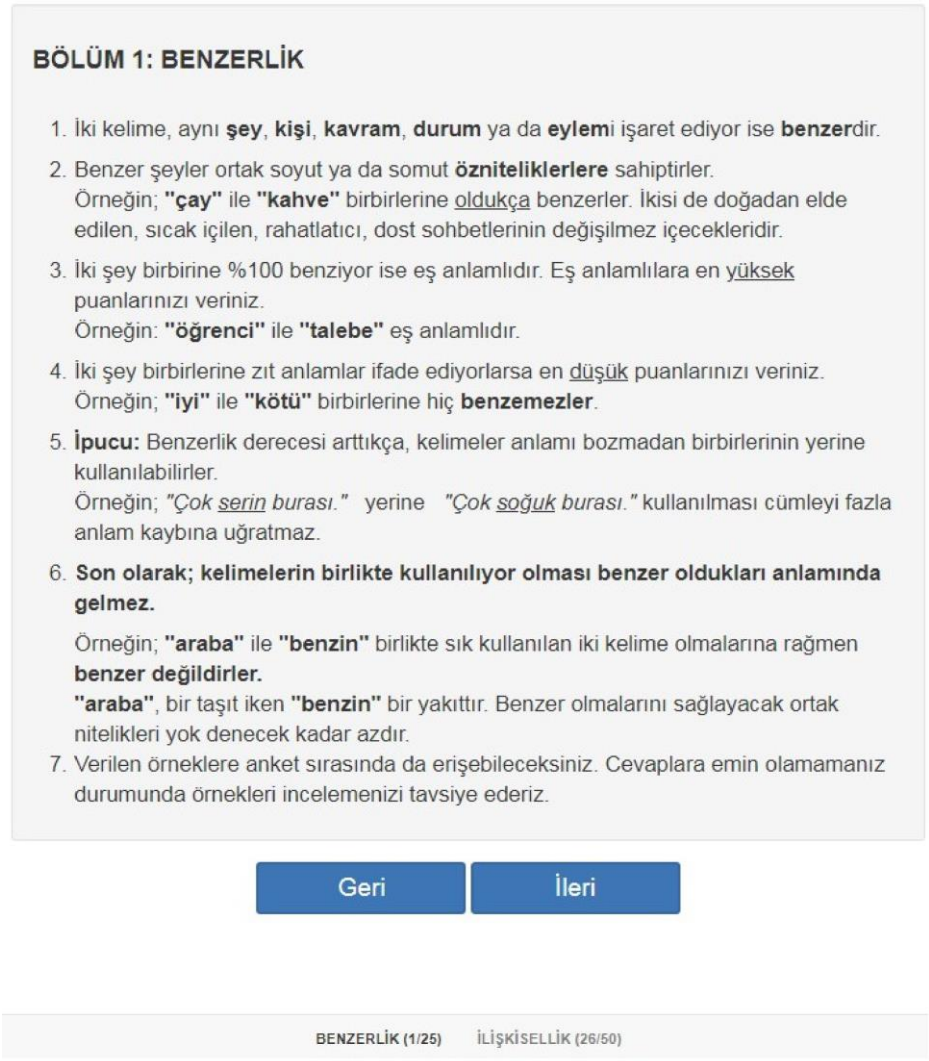

Figure 7.11 AnlamVer Questionnaire - Similarity Definition Screen

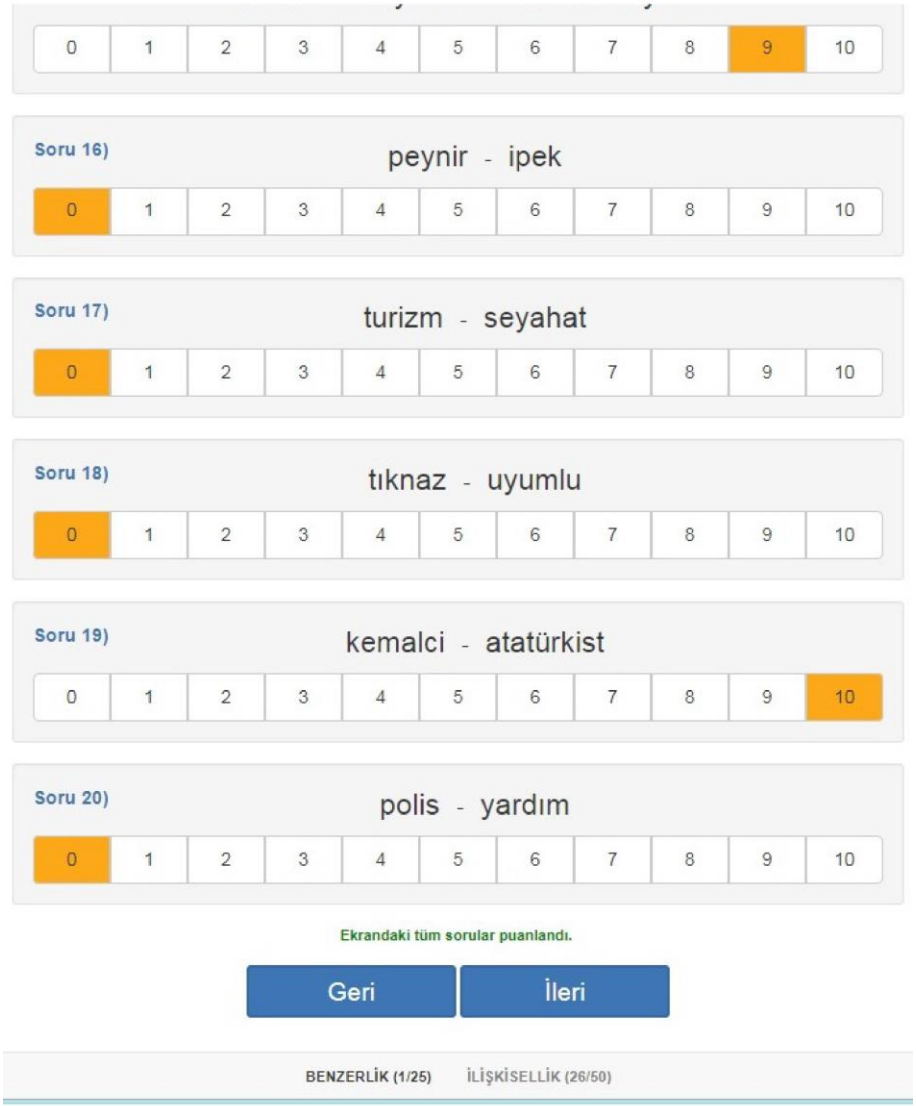

Figure 7.12 AnlamVer Questionnaire - Similarity Annotation Screen

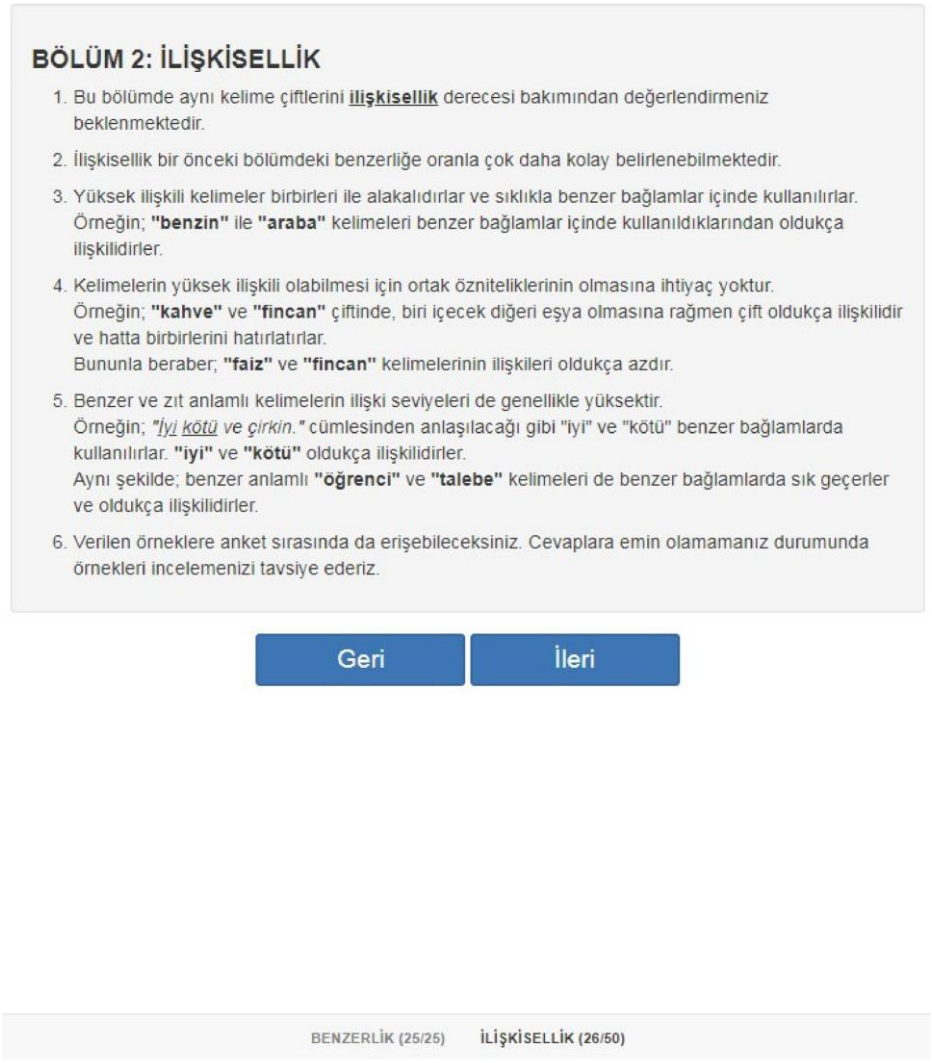

Figure 7.13 AnlamVer Questionnaire – Relatedness Definition Screen

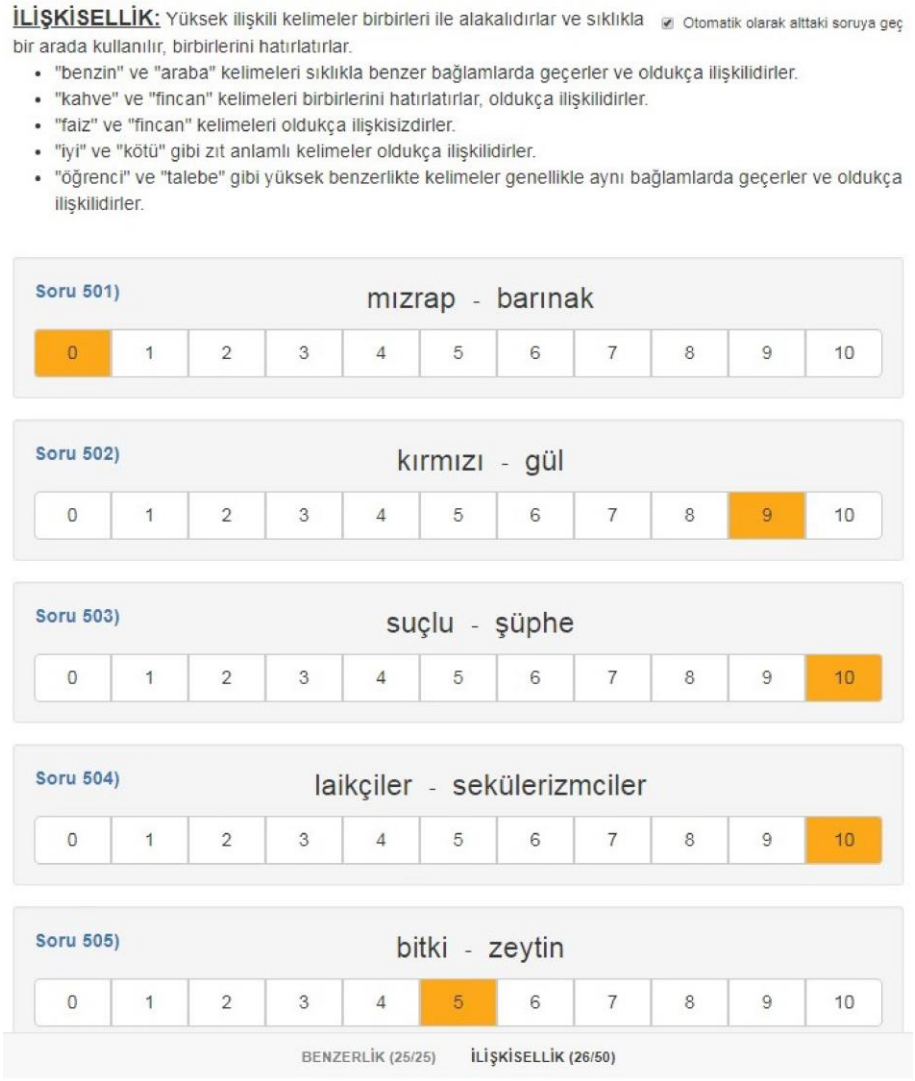

Figure 7.14 AnlamVer Questionnaire – Relatedness Annotation

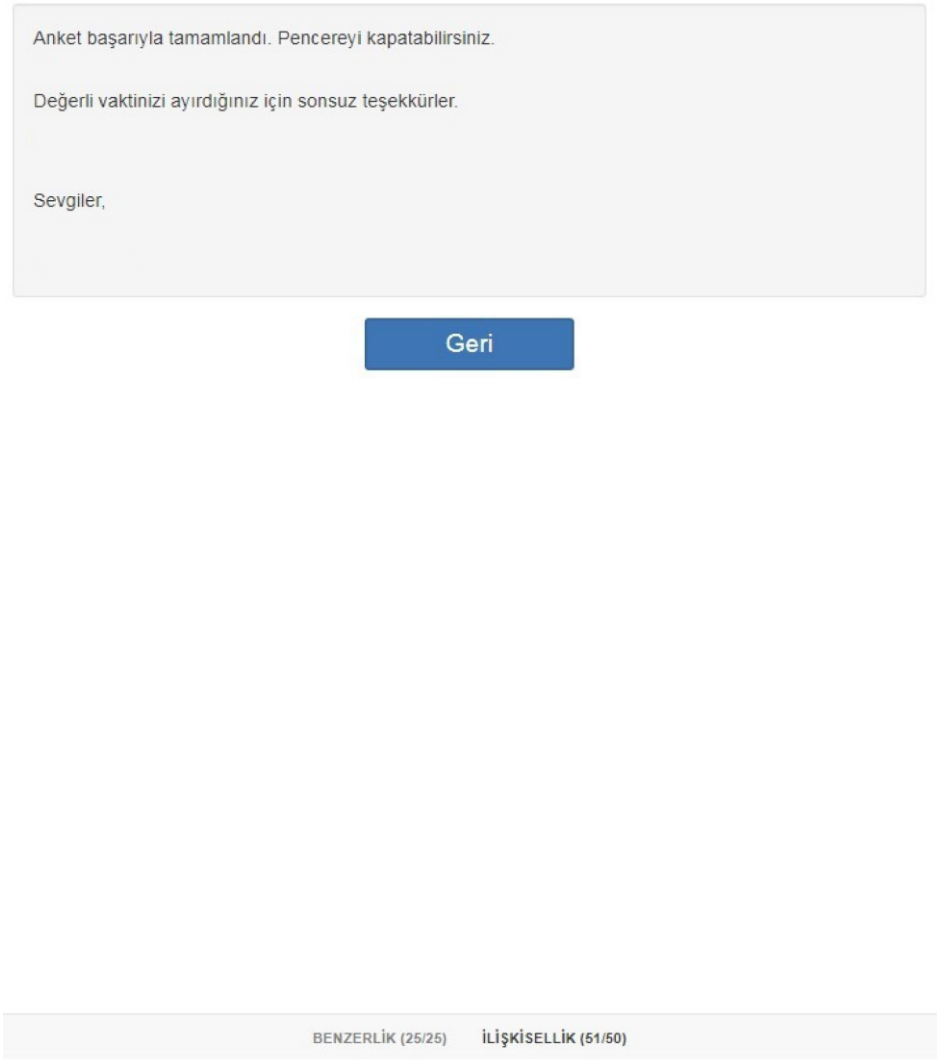

Figure 7.15 AnlamVer Questionnaire – End Screen

CURRICULUM VITAE