IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 January 2025, accepted 26 March 2025, date of publication 2 April 2025, date of current version 18 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3557086

RESEARCH ARTICLE

Grammar or Crammer? The Role of Morphology in Distinguishing Orthographically Similar but Semantically Unrelated Words

GÖKHAN ERCAN[1] **AND OLCAY TANER YILDIZ**[2]
[1]Department of Computer Engineering, Işık University, 34398 İstanbul, Türkiye
[2]Department of Computer Engineering, Özyeğin University, 34794 İstanbul, Türkiye
Corresponding author: Gökhan Ercan (mail@gokhanercan.com)

ABSTRACT We show that n-gram-based distributional models fail to distinguish unrelated words due to the *noise* in semantic spaces. This issue remains hidden in conventional benchmarks but becomes more pronounced when orthographic similarity is high. To highlight this problem, we introduce OSimUnr, a dataset of nearly one million English and Turkish word-pairs that are *orthographically similar but semantically unrelated* (e.g., grammar – crammer). These pairs are generated through a graph-based WordNet approach and morphological resources. We define two evaluation tasks—unrelatedness identification and relatedness classification—to test semantic models. Our experiments reveal that FastText, with default n-gram segmentation, performs poorly (below 5% accuracy) in identifying unrelated words. However, morphological segmentation overcomes this issue, boosting accuracy to 68% (English) and 71% (Turkish) without compromising performance on standard benchmarks (RareWords, MTurk771, MEN, AnlamVer). Furthermore, our results suggest that even state-of-the-art LLMs, including Llama 3.3 and GPT-4o-mini, may exhibit noise in their semantic spaces, particularly in highly synthetic languages such as Turkish. To ensure dataset quality, we leverage WordNet, MorphoLex, and NLTK, covering fully derivational morphology supporting atomic roots (e.g., '-co_here+ance+y' for 'coherency'), with 405 affixes in Turkish and 467 in English.

INDEX TERMS Derivational morphology, distributional semantic modeling, language resource, morphological segmentation, orthographic similarity, word-relatedness, word-similarity.

I. INTRODUCTION

Subword-level modeling has gained popularity in natural language processing (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 [1] represents ≈70K words with ≈15K roots (including some proper nouns) and 422 affixes—this concept initially sounds appealing. Assuming Zipf's law [2] 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 [3] 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

The associate editor coordinating the review of this manuscript and approving it for publication was Ajit Khosla.

systematic compositionality of languages is questionable,'' studies suggest that deep networks are capable of making subtle grammar-dependent generalizations'' [4]. 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.

A. WORD SEGMENTATION

Subword-level distributed semantic modeling (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) [5] 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 [6], Byte Pair Encoding (BPE) [7], SentencePiece [8] or supervised segmentation methods such as CHIPMUNK [9]. 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. [10], 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 Williams et al. [11], *phrases* obey Zipf's law more closely than words and other subword units, and they comprise the most coherent units of meaning.

B. 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 [12], 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 [13] 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 [14], [15] 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.[1,2] On the contrary, as Bender [16] 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. More broadly, as Sutton [17] 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.

C. THE ROLE OF 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 [12], [18], [19], [20]. A study by Zhao et al. [19] shows that the statistical methods, such as BPE and Morfessor, cannot outperform the FastText n-gram benchmark on the Turkish word relatedness dataset AnlamVer [21]. They state that it is due to the *noise* generated from the syntactic affix concatenations in Turkish. Another study [20] reports that'' the choice of subword units—morphemes or n-grams—doesn't make much difference'' on part-of-speech (i.e., POS), Chunking, and NER tasks in Russian language. An earlier study [18] 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

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

TABLE 1. 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 (§IV-C). FT: default FastText, FT-M: morphological segmentation, FT-MR: root-only morphological.

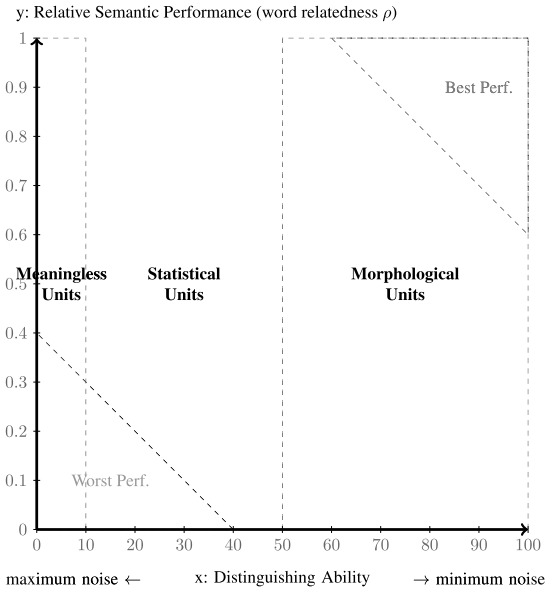

FIGURE 1. Semantic Clarity Space: A conceptual diagram illustrating how noise decreases as the meaninglessness of segmentation units decreases. Refer to §VI-D4, Table 24, and Figs. 13 and 14 for formulation and empirical results.

compared to Morfessor and syllable segmentation (e.g., Root=43.29, Morfessor=40.32, Syllable=41.29). In another work, Üstün et al. [22] 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 4 and Fig. 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 22).

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 [23], MMLU [24] or BIG-Bench Hard [25], 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).

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 Fig. 1). 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 [26], *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 [27] 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 18). Conversely, morphological segmentation overcomes the problem (en=68%, tr=71% accuracy) while performing similarly on conventional wordsim evaluations (Table 22, Fig. 13). Table 1 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).

D. CONTRIBUTIONS

The main contributions of this paper are as follows: (i) construction of a publicly available[3] 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, (ii) development of an open-source,[4] extensible dataset construction tool, including orthographic similarity and WordNet algorithms, and an English morphology stack. (iii) 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, (iv) proposal of unrelatedness-identification and relatedness-classification tasks, which provides insights into measuring the distinguishing ability of models, and experimentation with the task using various word segmentation settings, (v) benchmarks of WordNet-based relatedness/similarity approximation algorithms on word similarity datasets and the proposed task, (vi) development of a methodology on applying fully derivational morphology (reducing to atomic roots) for English and Turkish by mixing both human-annotated resources and real-time morphological analysis and disambiguation tools.

II. BACKGROUND AND MOTIVATION

A. 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'' [28]. 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 [29]. 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* [30] 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 [29], AnlamVer [21], SuperSim [31], SimRelUz [32] 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 [10], [31], [33], 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).[5] In their studies, Hengchen and Tahmasebi [31] and Hill et al. [29] conclude that modeling the relatedness is easier than modeling the similarity. Our word relatedness and similarity experiment (Table 22) also justifies this hypothesis on Turkish AnlamVer dataset ($\rho_{sim} = 0.44$, $\rho_{rel} = 0.74$), 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 [21], [34], [35]. 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 [36]. 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 this study, 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

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

[4] https://github.com/gokhanercan/OSimUnr-Generator or http://gokhanercan.com/OSimUnr-Generator

[5] AnlamVer Turkish dataset includes two distinct scores for each word-pair which here referred to AnlamVerSim for similarity and AnlamVerRel for relatedness.

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 experiments for benchmarking purposes only. Therefore, we exclude them from aggregate results of the relatedness experiments (Table 22, 21).

B. THE NOISE

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])[6] segments the word *glowing* into 22 grams such as *'glo, glow, glowi, …, win, low, lowi, …, wing, ing'* as shown in Table 2. 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-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. [11], 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 2).

[6]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*).

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 2, 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). 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 [37]. When the edits are in the middle, and the word lengths are short, the overlap is relatively lower (22.22% for *fri<u>d</u>ge – fri<u>n</u>ge*). But when the one edit distance is on the first or last character, for highly derivational and/or inflectional cases like *<u>t</u>encerelerimizden – <u>p</u>encerelerimizden*, 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.

3) 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 $\rho = 0.50$) between FastText's predictions and the orthographic similarity scores of orthographically-similar word-pairs. Fig. 2 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

TABLE 2. An example from overlapping units of Char-gram[3-6] Segmentation.

Word	Char-gram[3,6] Segmentation	Morphological Segmentation
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.

TABLE 3. Linear relationship between word lengths (Len) and Char-gram[3-6] (CG) Overlaps (OL). All edits are in the first letter of words. See Equation 3 for editsim formulation. FT and FT-MR relatedness scores are normalized to [0-1] scale.

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

model exhibits no correlation with orthographic similarity algorithms.

C. DATASET REPRESENTATION PROBLEMS

Irrespective of whether they measure similarity or relatedness, conventional wordsim datasets (i.e. the wordsim task) such as WordSim353 [38], RG [39], MG [40] have long served as one of the main performance measures for DSMs alongside the analogy task [41]. 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 [42], [43], [44], 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 [42]:" 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.

1) TASKS MEASURE RELATIVE RELATIONSHIPS, NOT ABSOLUTES

The wordsim task simply measures the ρ Spearman rank correlation [45] 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* (Fig. 9 and 10), 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 ($\rho_{rel} = 0.61$, $\rho_{RG} = 0.77$, Table 22). 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 1).

2) DISTRIBUTIONAL MISMATCH

Since there is no consistent methodology for selecting word-pools and word-pairs for the construction stage of wordsim datasets [44], datasets tend to vary in relation types, POS constraints, word frequencies, morphological forms

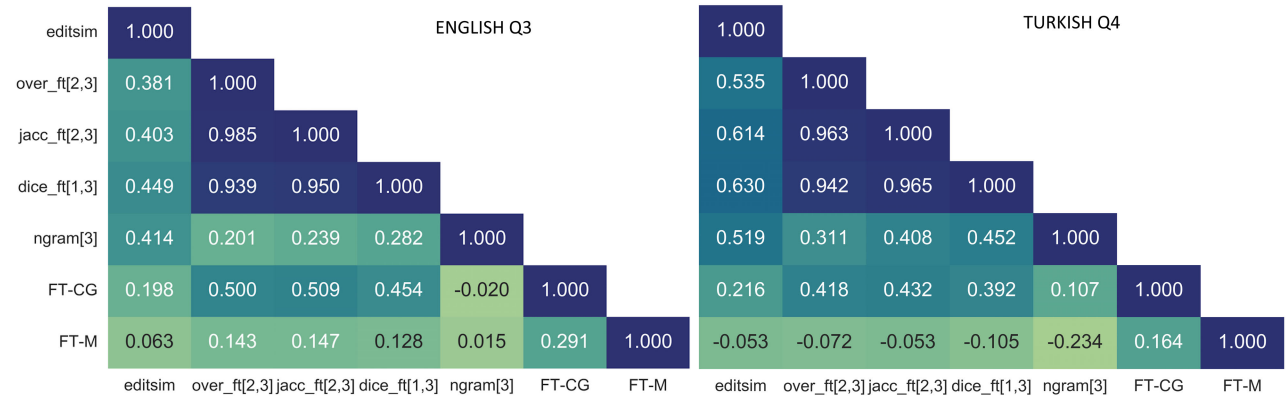

FIGURE 2. Spearman correlations (ρ) of similarity scores for semantic models and orthographic similarity algorithms. Experiments ran on OSimUnr editsim dataset.Q3 word-pairs are sampled to 20,000 items.Dice, Jaccard, Overlap coefficients are calculated using FastText n-grams. Ngram[3] (i.e., ngr3) denotes n-gram similarity algorithm. See §II-F for the algorithms.

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. [44], among the 51 datasets it covers, MEN [46] is the largest word relatedness dataset for English, containing only 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.

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 [47], authors tend to choose words that are related (e.g., *brother – lad*) during the word-pairing stages, shown in Table 4 and Fig. 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 4 and Fig. 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 4), 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. 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.

TABLE 4. Average orthographic similarities and lengths of some existing wordsim datasets.

Dataset	Type	Scale	Size	Len	Score	ES	NG
MC [40]	rel	0-4	30	5.50	4.92	1.21	0.96
RG [39]	rel	0-4	65	5.80	4.69	1.13	0.89
WS353 [38]	rel	0-10	353	6.49	5.86	1.42	1.25
RareWords [48]	rel	0-10	2,034	8.75	6.21	2.24	1.94
MEN [46]	rel	0-50	3,000	5.50	5.00	1.32	1.17
MTurk771 [49]	rel	1-5	771	6.14	7.4	1.29	1.09
SimLex-999 [29]	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 [21]	both	0-10	500	6.78	4.98	1.53	1.47
AnlamVerOOV* [21]	both	0-10	66	10.98	4.73	2.77	2.50
Sopaoglu [50]	rel	0-5	101	5.60	5.75	0.98	1.06
WordSimTr [22]	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

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

D. 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

TABLE 5. 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) [51].

of synthesis [51]) 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 5). 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.

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,[7] 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.

E. ORTHOGRAPHIC SIMILARITY - RELATEDNESS SPACE

To illustrate the main focus of this study, OSimUnr word-pairs, we define OSIM-REL space (Fig. 3) following the idea of SIM-REL space from the AnlamVer study [21]. 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 ($x = rel(w_1, w_2), y = sim(w_1, w_2)$). 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 ($sim(w_1, w_2) > 7.5, rel(w_1, w_2) > 7.5$) (e.g., *car – automobile*), or *antonyms* if their relatedness is high but similarity is low (e.g., *hard – easy*).

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 $[w_1, w_2]$, we calculate $OSim(w_1, w_2)$ 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 ($x = Rel(w_1, w_2)$) should be obtained from existing relatedness datasets, DSMs such as FastText, or WordNet-based relatedness/similarity approximation algorithms, which we cover in §IV. We define t_x and t_y ($0 < t_x < 5, \quad 0 < t_y < 5$) as threshold

[7]Essential #22: Languages vary in how easy it is to find the boundaries between morphemes within word [16].

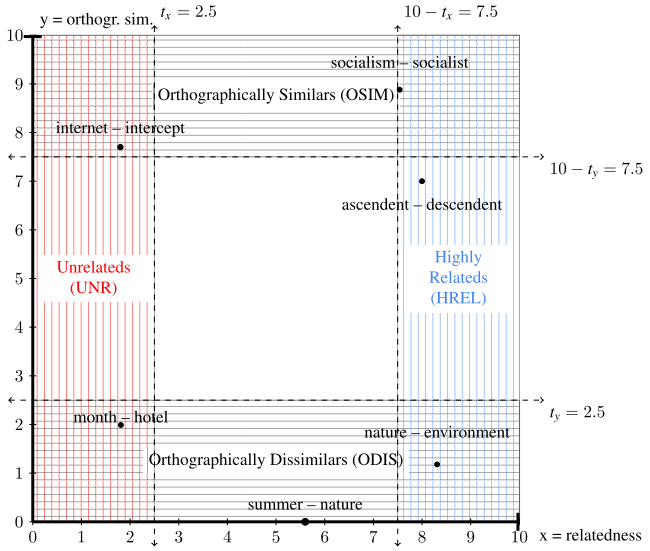

FIGURE 3. OSIM-REL: Orthographic Similarity - Relatedness Space of Word-pairs.Threshold variables t_x, t_y equally selected as 2.5. Unrelateds (UNR) area is vertically hatched in red while Highly Relateds (HREL) area hatched in blue. Orthographically Similars (OSIM) and Orthographically Dissimilars (ODIS) areas in horizontal black lines.

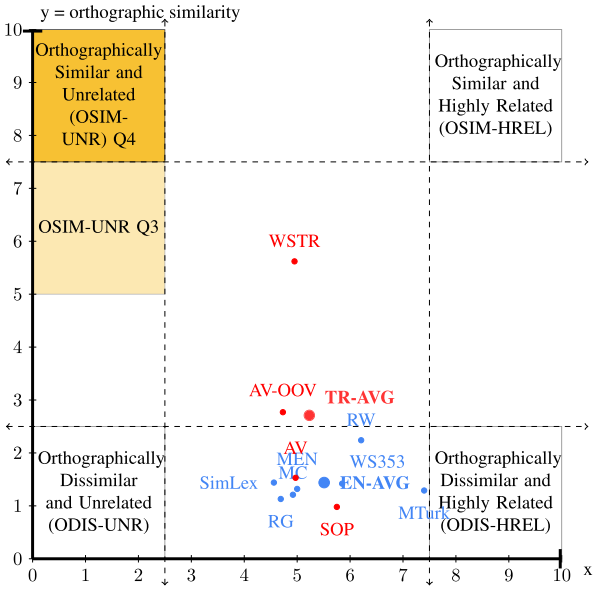

FIGURE 4. Sub-regions of OSIM-REL Space. Points represents average scores of wordsim datasets (RW: Rarewords, SOP: Sopaoglu, AV: AnlamVer, WSTR: WordSimTR). Bold points denote average wordsim score for each language (EN-AVG, TR-AVG). Red and blue dots denote Turkish and English datasets. Area in yellows (OSIM-UNR) are the main focus of this study. All dataset scores are normalized to [0-10] scale.

variables that determine decision boundaries for x and y axes respectively. Fig. 3 illustrates how the conceptual OSIM-REL space defines *ss* sub-spaces with the function f_1 (Eq. 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 Fig. 3, and the sub-regions such as OSIM-UNR Q3/Q4 in Fig. 4, illustrate the basic assumptions of this study in determining degrees of orthographic similarity and relatedness measures.

$$ss = f_1(w_1, w_2) = (x = Rel(w_1, w_2),$$
$$y = OSim(w_1, w_2), t_x, t_y) = \begin{cases} \text{OSIM}, & \text{if } y \geq 10 - t_y \\ \text{ODIS}, & \text{if } y < t_y \\ \text{UNR}, & \text{if } x < t_x \\ \text{HREL}, & \text{if } x \geq 10 - t_x \end{cases} \quad (1)$$

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 (Eq. 2). As Fig. 4 highlights in yellow, *orthographically-similar-but-unrelated* (OSIM-UNR) *Q3* and *Q4* sub-regions are the main focus of this study. We add the Q3 sub-region into our dataset to have more word-pairs (more than 99% of all word-pairs, Table 10) and to be able to measure the contribution of orthographic similarity to performance (see left-to-right trend in Fig. 11). Fig. 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.

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

F. 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 Eq. 3.

$$editsim = y = OSim(w_1, w_2)$$
$$= 1 - NormalizedEditDistance(w_1, w_2) \quad (3)$$

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 6). 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 6, 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 [52], from consideration.

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 [53], 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 2), which places higher value on the beginning n-grams by adding beginning characters ('<') to words before generating n-grams.[8] 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 6, 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* (i.e., Szymkiewicz – Simpson coefficient) 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 (Eq. 4). This coefficient provides a measure of similarity that accounts for overlapping segments between words.

$$overlap(seg_{w1}, seg_{w2}) = \frac{|seg_{w1} \cap seg_{w2}|}{min(|seg_{w1}|, |seg_{w2}|)} \quad (4)$$

The overlap coefficient (overft* columns in Table 6) 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 6.

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 study. 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 18). As anticipated, the final `editsim` sub-dataset contains more word-pairs (≈570K) than the final `over_ft23` sub-dataset (≈70K), as shown in Table 10. 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, FT-MR on, `over_ft23` (64.82%) compared to `editsim` (68.47%). For most algorithm implementations, we utilize the python-string-similarity package.[9] To enhance runtime

[8] Square brackets in "ft[2-3]" indicate beginning and ending characters are included in the n-grams.

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

TABLE 6. 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

performance, we cythonize [54] the library, meaning that converting it to its C programming-language equivalents.

III. DERIVATIONAL MORPHOLOGY

A. ASSUMPTIONS ON MORPHOLOGY AND LANGUAGE

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 *-prefix$_1$...-prefix$_p$_root$_1$..._root$_r$+suffix$_1$+...+suffix$_s$* format for morphological segmentations (e.g., *-co_here+ance+y* for *coherency*).

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).[10] English is a fusional language; therefore, it supports compounding of words, which can form multiple root morphemes per word (e.g., *_dog_house* for *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)*.

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.

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 weather 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) 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*.[11] 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 [16].

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,[12] 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.

B. MODELING DERIVATIONAL MORPHOLOGY

In this study, we utilize morphological information for two distinct purposes: a) to facilitate automatic dataset generation

[10] Essential #11: Root morphemes convey core lexical meaning [16].

[11] Essential #12: Derivational affixes can change the lexical meaning [16]. Example from the book.

[12] Essential #14: Inflectional affixes add syntactically or semantically relevant features [16].

by detecting shared roots, and b) to model atomic sub-units of language for training.

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" (§III-A2), we consider two words to be related if we identify that they share at least one of 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.

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 [1], 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)}"[13] 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.

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 [55]. 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 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 *acceptability* and *acceptable*. 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 [6] is a supervised model trained using the Conditional Random Field (CRF) 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 7), 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 [56]. Another method we employ utilizes the *derivationally-related-form* association of lemmas from WordNet (WN column in Table 7). 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 (§ IV-D2), rather than the morphology stack.

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.

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

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

TABLE 7. Hand-picked examples from shared root detection experiments for English.

Word-pair	Morphy	WN	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]

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 denote that WordNet has prior knowledge that two words are derivationally-related without knowing the actual roots. *Full Stack combines Morphy, MorphoLex, and WordNet (with its word-pool only) with shallow affixations. Morfessor2 is not included in the English stack.

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 $\mathcal{O}(R + S + A)$ 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 $\mathcal{O}(QueryWords \times (R + S + A))$.

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 [57], 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. [58] 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 [58],[14] 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 8 shows sample lexicon and suffixation rule definitions from our implementation.

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 Fig. 5, 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 [59]. However, it should be noted that this alignment is not always the case, and when a static analysis is available, it is preferred.

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 correspond-

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

TABLE 8. 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, ^DB: Derivation.

Lexicon (txt file)	Suffixation Rules (xml 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 ..	<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>

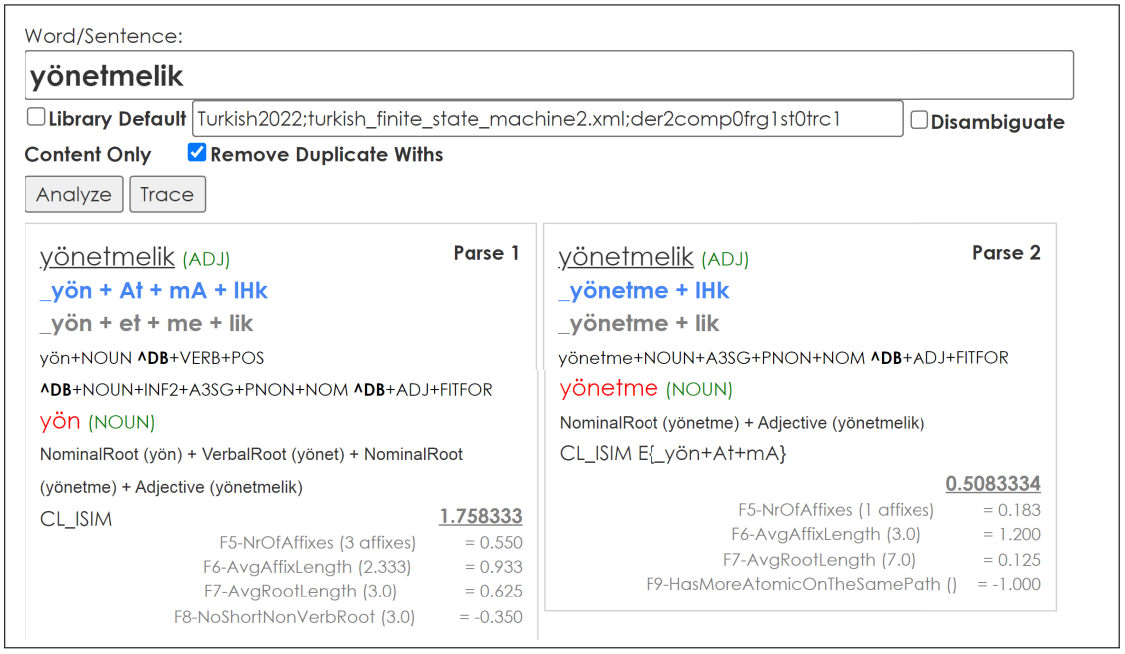

FIGURE 5. Example of an Atomic Morphological Analysis with Disambiguation Scores. The screenshot is taken from our morphological analysis and disambiguation user interface implementation.

ing morphological tags: "yön+NOUN ^DB+VERB+POS ^DB+NOUN+INF2+A3SG+PNON+NOM ^DB+ADJ+FIT FOR". 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 8). 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 paper. 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.

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 [59] 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* 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.

TABLE 9. Four main stages of the dataset construction pipeline.

#	Stage	Input Type	Output T.	Output Sample
1	Word-pool Selection	[Word src.]	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

IV. DATASET CONSTRUCTION

We designed a dataset construction pipeline for automatically building OSimUnr word-pairs in four main stages (Table 9). 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.[15] Our dataset construction pipeline does not contain human intervention, except for the sub-stage Categorical Filters (see §IV-D3). 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.

A. WORD-POOL SELECTION

The first stage is word-pool selection, which aims to automatically select word candidates from existing resources based

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

on certain word filtering constraints, rather than manually hand-picking them. As initial word sources for the pipeline, we employ WordNet 3.0 [60] through the Python implementation NLTK [55] for English. For Turkish, we utilize WordNet KeNet [61] along with its Java implementation.[16] 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 [62]; 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, 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 [62] 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.

Another constraint we applied to word-pools is the minimum word length. As our analysis (see §II-B1) 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 10).

B. 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 (Eq. 3). Since the complexity of the matching process is quadratic ($\mathcal{O}((n/2)^2)$), it would normally take about a week to execute matchings per language on a standard computer.[17] 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 ($5 <= OSim(w_1, w_2) <= 7.5$) 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 10). 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 derivational morphology at the fourth stage of the pipeline, Relatedness Filtering (§IV-D).

C. WordNet RELATEDNESS APPROXIMATION

The previous stage yields millions of word-pairs ($\approx$ 5.7M for English, $\approx$ 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* or *unrelated*. 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.

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

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

TABLE 10. 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.

Stage	Dataset Construction Stages	English		Turkish	
1	**Word-pool Selection**				
	Word-pool Source (reference)	English WordNet 3.0 [60]		Turkish WordNet KeNet [61]	
	WordNet Implementation	NLTK [55]		Java Lib.[16]	
	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

1) APPROXIMATION METHODS

We employ six (three for Turkish) WordNet-based methods at our disposal: `wup` [63], `path` [62], `lch` [64], `lin` [65], `jcn` [66], `res` [67]. These methods are often referred to as *similarity* measures [62] rather than relatedness [68], [69]. These methods define path distance based formulations to approximate similarity/relatedness by incorporating IS-A relationship nodes (synsets) of WordNet databases (Eq. 5,6). 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). 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).

$$wup(c1, c2) = 2 \times \frac{depth(lcs(c1, c2))}{depth(c1) + depth(c2)} \quad (5)$$

$$lch(c1, c2) = -log\frac{len(c1, c2)}{maxdepth(c), c \in WordNet} \quad (6)$$

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 Fig. 6, 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` [70] could potentially provide better results (see *no-relations* line in the figure). For example, more comprehensive lexical resources such as Concept.Net [71] 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 §II-A), 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.

FIGURE 6. WordNet IS-A type graph depicts how related concepts can be distant in path distance.

In the AnlamVer study [21], 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 Fig. 6. To address this weakness, we introduce additional relatedness detection pipelines in the subsequent stages, leveraging type hierarchy and morphology.

2) 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. [72] 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 25 in Appendix A, 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. [69] 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, Sopaoğlu and Ercan [50] measured relatedness using three WordNet-based methods. We refer to their dataset as Sopaoglu (see Table 4), 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 Sopaoğlu and Ercan [50].

TABLE 11. WordNet relatedness approximation experiments measured by relatedness-classification and word relatedness tasks.

	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	-	-	-

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. See the full version of this table in Appendix 25.

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 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 11). 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 11.

In addition, Table 25 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 22) 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.

D. 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 12 displays the sub-stages of the pipeline.

TABLE 12. Stage 4: Relatedness filtering sub-stages.

# 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 entry 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 Affixes	cyberart – cyberwar	*-cyber* adds its own meaning, it is in the affix blacklist.

1) SHARED ROOT DETECTION

Within the Morphology stack (§III-B1), 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.

2) SEMANTIC FILTERS

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

a: 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 11). 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.

b: DERIVATIONALLY-RELATED FILTER

Following the assumption that *words derived from the same root are related* (see §III-A2), 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

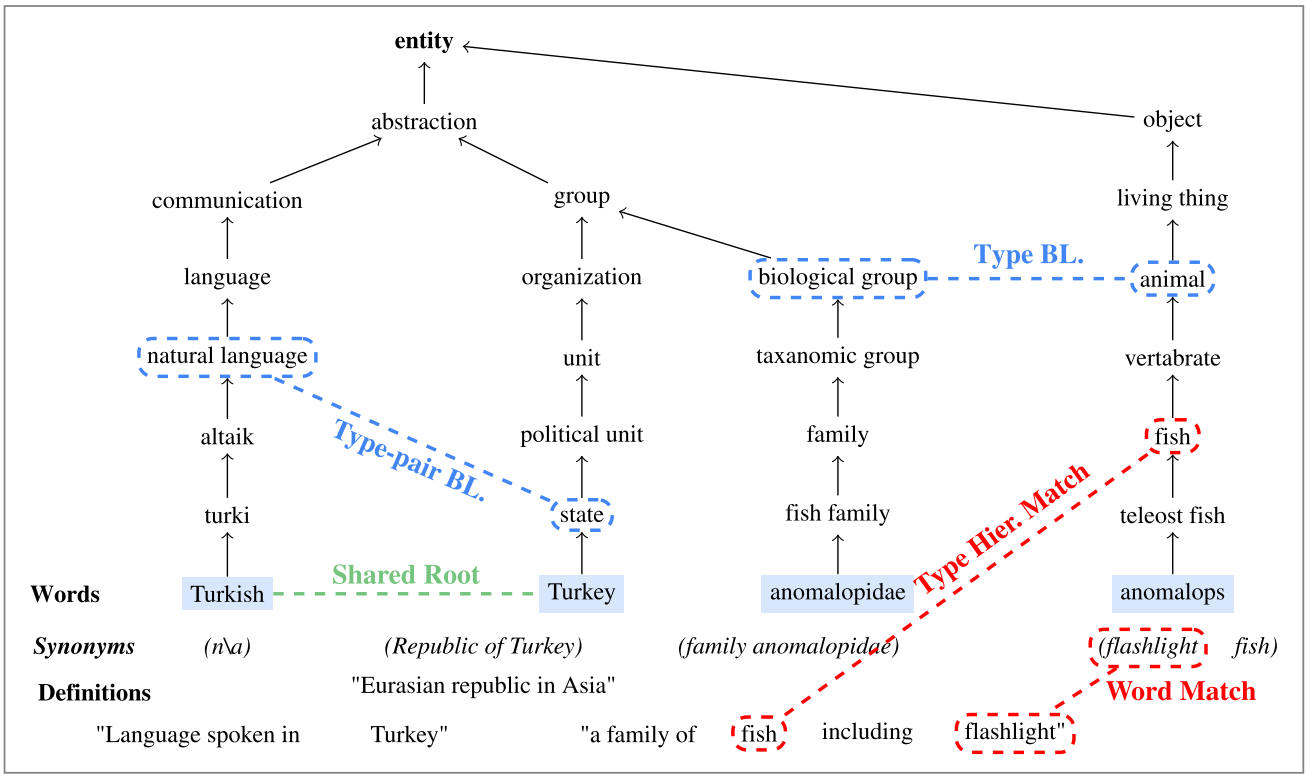

FIGURE 7. Simplified examples demonstrating filter types on WordNet type graph. Some concepts are omitted in the hierarchy for clarity. Definitions are shortened and changed slightly for clarity.

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.

c: 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 Fig. 7, 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*.

d: 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 Fig. 7, 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.

3) 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

TABLE 13. Essential 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 minOrthographic-SimQ4. 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.

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.

a: 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 Fig. 7, 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.

b: 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

```
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 8. Simplified Abstract PipelineProviderBase Class.

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 Fig. 7, 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 of them shown in Fig. 7) 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*).

c: COMMON MEANINGFUL AFFIXES

As discussed by Bender [16], 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.

E. REPRODUCIBILITY AND LANGUAGE RESOURCES

We open-source the Python implementation of the dataset generation pipeline, named OSimUnr-Generator,[18] 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.

1) ASSUMPTIONS AND PARAMETERS

Based on the OSIM-REL space definition (Fig. 4, Eqs. 1 and 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 13.

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)
```

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

EnglishPipeline is the default concrete provider implements the following factory methods of the PipelineProviderBase class (Fig. 8), 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 (IV-D3). 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.

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 [73] and MorphoLex [1]. 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 10). Similarly, the primary components of the English[19] 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 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 [58], which was customized for this study to support derivational morphology with an atomic roots lexicon. As discussed in Section II-D, 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.

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 [58]), the number of universal databases

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

and analyzers remains very limited. It appears that current resource landscape aligns with Bender's statement [16]: *"...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 14, 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.

a: OFF-THE-SHELF RESOURCE IMPLEMENTATIONS

The first row of the Table 14 highlights French (fra) as the only fully implementation-ready resource, aside from Turkish and English, as it has a MorphoLex-fr [74] 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,[20] 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 [75].

Table 14 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 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.

b: UNIMORPH 4

UniMorph [76], 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 UNESCO. 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 (Section III-B4), 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 [77] 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 [78], a multilingual morphological database with 519,000 derivational and 10.1 million inflectional entries.

c: CONCEPTNET 5.5

For relatedness approximation, one alternative universal resource is ConceptNet [71], an open multilingual knowledge graph representing 304 languages,[21] 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 [79]. 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 14 are grouped based on ConceptNet vocabulary size categories and the availability of lexical resources, filtered to include only those that satisfy both criteria.

[20] https://omwn.org

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

TABLE 14. Resource availability for new language adaptation.

Resource Availability	Voc#	Forms#	Rel. Approx.	Lexicon	#	Languages (ISO 639-3)
Fully Implemented	55,350	38,340	NLTK WordNet	MorphoLex-fr	3	**eng**, ***tur***, **fra**
Requires UniMorph Impl.	20,000	50,000	NLTK WordNet	UniMorph 4	8	**cat**, ***fin***, **ita**, nld, **pol**, **por**, slv, **spa**
	20,000	10,000	NLTK WordNet	UniMorph 4	3	*eus*, glg, ind
	3,000	10,000	NLTK WordNet	UniMorph 4	10	ara, bul, dan, ell, fas, heb, nno, nob, sqi, **swe**
Requires Two Impls.	20,000	50,000	ConceptNet 5.5	UniMorph 4	17	fro, **ger**, gle, hin, **hbs**, ***hun***, hye, isl, *kat*, **lat**, lav, mkd, ron, **rus**, slk, sme, xcl
	20,000	10,000	ConceptNet 5.5	UniMorph 4	8	ast, bel, **ces**, *est*, grc, *kaz*, lit, ukr
	3,000	10,000	ConceptNet 5.5	UniMorph 4	19	*ady*, afr, ang, cym, dsb, fao, frm, *guj*, *nav*, oci, osx, *que*, *sah*, san, syc, urd, *uzb*, vec, *vep*
	0	0	ConceptNet 5.5	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.

V. EXPERIMENT SETUP

A. EXPERIMENTS

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 18 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.

2) EXPERIMENT 2 - WORD RELATEDNESS

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 22 and 21). This task ensures that we do not compromise performance on an existing *relative task* while improving our primary objective of distinguishing ability (Table 18). 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 Fig. 1 and 13).

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 20). 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.

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.

a: WORDSIMS

The first sub-dataset, WordSims,[22] is a combined version of all relatedness datasets used in this study (Table 4). It includes 6,170 word-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 22,21). In this experiment, and

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

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.

b: OSIMBINARY

The second sub-dataset, OSimBinary,[23] was created using the generator pipeline (Stage 4 in Table 10). 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 12), 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), whereas the WordSims dataset consists of manually curated pairs and is biased toward frequent words, as explained in Section II-C2. These characteristics make this sub-dataset the most challenging element in our experiments.

B. MEASURES

Aside from the traditional Spearman ranking correlation p measure of the word-relatedness task (Table 22), we also utilize the following measures:

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 $IsUnrelated_m(w_1, w_2)$ function to continuous model (m) predictions we get from the $Rel_m(w_1, w_2)$ function (Eq. 7 and 8).

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

As explained in Section IV-D1, the ground truth labels of this task are *unrelated* OSimUnr word-pairs which are achieved by applying the same threshold function $IsUnrelated_{wn}(w_1, w_2)$ to normalized WordNet (*wn*) relatedness approximations ($Rel_{wn}(w_1, w_2)$). 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 (Eq. 9).

$$IsUnrelated_m(w_1, w_2) = Rel_m(w_1, w_2) < t_x \tag{7}$$

$$TruePrediction_m(w_1, w_2) = IsUnrelated_m(w_1, w_2) = IsUnrelated_{wn}(w_1, w_2) \tag{8}$$

$$acc = (TP + TN)/(TP + FP + TN + FN) \tag{9}$$

$$pre = TP/(TP + FP), \quad rec = TP/(TP + FN) \tag{10}$$

$$F_1 = 2 \cdot pre \cdot rec/(pre + rec) \tag{11}$$

$$Specificity = TN/(TN + FP) \tag{12}$$

2) RECALL, PRECISION AND F_1 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 19) using the standard precision, recall, and F_1 measures, as defined in Eqs. 10,11. 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. 12), 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.

3) MEAN ABSOLUTE ERROR (ERR)

Since our main tasks are to distinguish concepts from each other, we inevitably applied *hard thresholding* using the "IsUnrelated function" (Eq. 7) 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, Fig. 9 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 Fig. 9 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 (Eq. 13). Although this measure has not been widely used in DSM evaluation, it holds value as it provides an intrinsic benchmark for different model configurations. For example, in their study focused on measuring compositionality, Lazaridou et al. [80], 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 22).

$$err(w_1, w_2) = mae(w_1, w_2) = |y - Rel_m(w_1, w_2)| \quad (13)$$

C. 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 15, 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 [56], 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 [81], has an average of 12 tokens per sentence. Despite adding the OpenSubtitles2018 Corpus [82], 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 [83] and PolyglotWikiTR13 [56]. Although they were compiled using different extraction techniques in different years, there is a possibility of overlap between them.

TABLE 15. Corpora Utilized in Experiments. Voc: Vocabulary size (unique tokens), Sent: Number of sentences, Tok: Number of tokens, M: millions, B: billions. The Turkish corpus is a union of four separate corpora.

Corpus	Source(s)	Domain	Sent.	Voc.	Tok.
English	PolyglotWikiEN13 [56]	Encyclopedia	70M	5.5M	1.51B
Turkish	BounWebCorpus [81] ∪	News, Websites	189M	5.2M	1.24B
	OpenSubtitles2018 [82] ∪	Movie Subtitles			
	PolyglotWikiTR13 [56] ∪	Encyclopedia			
	trwiki-67 [83]	Encyclopedia			

D. 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

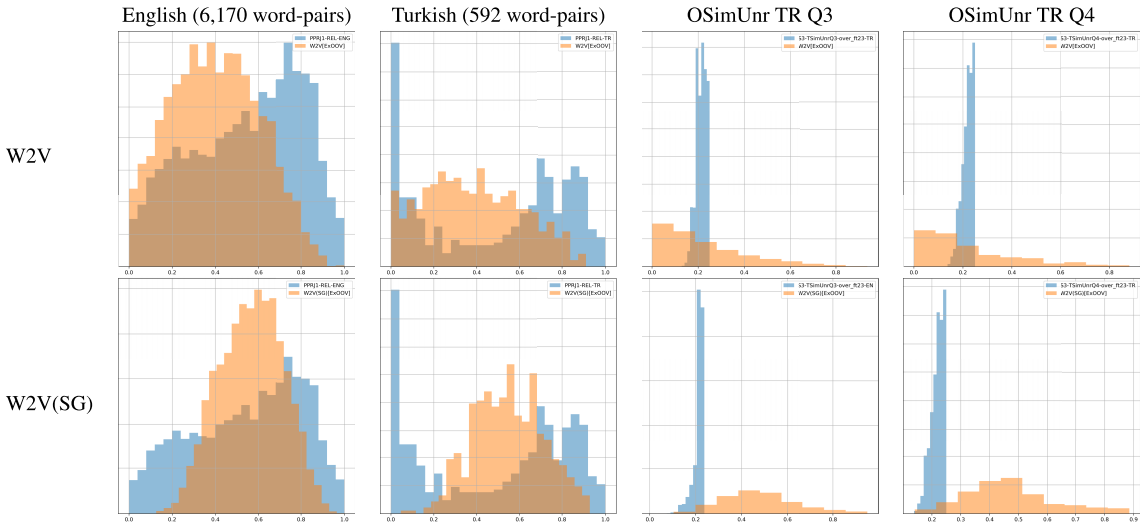

FIGURE 9. The histogram shows how relatedness distributes in a word-level SkipGram semantic space. Blue areas represent ground-truth word relatedness scores, while orange areas represent model predictions. Browns areas indicate overlapping regions.

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 [84], 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.

E. 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 16, 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.

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 2).

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

TABLE 16. 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*

they may offer a middle ground in terms of the *distinguishing words* task performance. For English hyphenation, we utilize the pyphen[24] library, which relies on Hunspell hyphenation dictionaries. This library provides comprehensive hyphenation rules for English words, enabling accurate segmentation into syllables. LibreOffice[25] 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'' [5]. 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 relatively meaningful units. Consequently, we adopt the (1-1) configuration settings for hyphenation in our experiments.

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*. 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 §III-A5). 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 III-A4). 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.

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 *gözlükçülük* is reduced only to the root *_göz* (Eq. 14). 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.

$$\begin{aligned} \text{\_göz (eye)} &= \text{\_göz}\cancel{\text{+lHk+CH}}\ \text{(optician)} \\ &= \text{\_göz}\cancel{\text{+lAm+sAl}}\ \text{(observational)} \end{aligned} \tag{14}$$

F. 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 [85], representing a locally hosted model, and GPT-4o-mini [86], 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.

[24] https://pyphen.org
[25] https://www.libreoffice.org

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.

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 (CoT) reasoning.

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[26] 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 [86], 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[27] 0-shot task, it achieves a score of 76.9 for Swahili language, where the score for English is 93.9 [86]. 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,[28] and we achieved similar results in our preliminary experiments.

3) LLAMA

Llama is a family of source-available models built on a dense transformer architecture [87], designed to support multilinguality, coding, reasoning, and tool usage, while being optimized for both efficiency and scalability [85]. The first model of the Llama 3 family, released in April 2024, was pretrained on a 15 trillion multilingual token corpus [85], 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.

a: 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 17). 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 order of magnitude faster than those with the Llama 3.3 model (Table 17). 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 21).

[26] https://platform.openai.com/docs/models#gpt-4o-mini

[27] https://huggingface.co/datasets/ebayes/uhura-arc-easy

[28] GPT-4o - $2.50/million tokens; GPT-4o-mini - $0.150/million tokens

b: 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 21). We report this configuration only in Turkish experiments. Although Llama 3 TRP demonstrated good performance on the word relatedness task (Table 21), 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 18). 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.

c: 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.[29] 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 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.

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

TABLE 17. 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.

TABLE 18. 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.

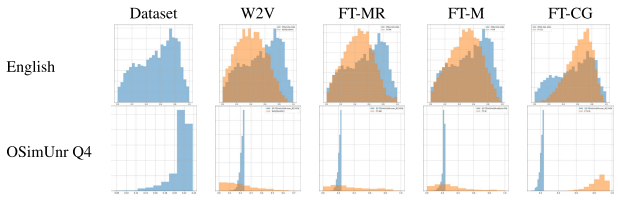

FIGURE 10. Histograms showing the relatedness distribution in CBOW semantic spaces using various segmentations. All distributions are given in the Appendix.

4) MODEL RUNTIME COMPARISON

We ran Llama models locally using the Ollama library[30] 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 17 presents the number of word-pairs that can be queried per minute to obtain relatedness scores for each

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

TABLE 19. 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.

TABLE 20. Experiment 3: Word-level Unrelatedness-identification Experiments on OSimUnr over_ft23 Datasets.

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

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. 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.

VI. RESULTS

A. RELATEDNESS CLASSIFICATION TASKS

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 18). 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 noise) values in the 'FT-CG Q4 *acc* cells' in Table 18. 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.

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 14 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.

a: WORDSIMS

Compared to unrelatedness-identification, this task is relatively more challenging because the WordSims dataset contains both related and unrelated scores, with 82% of

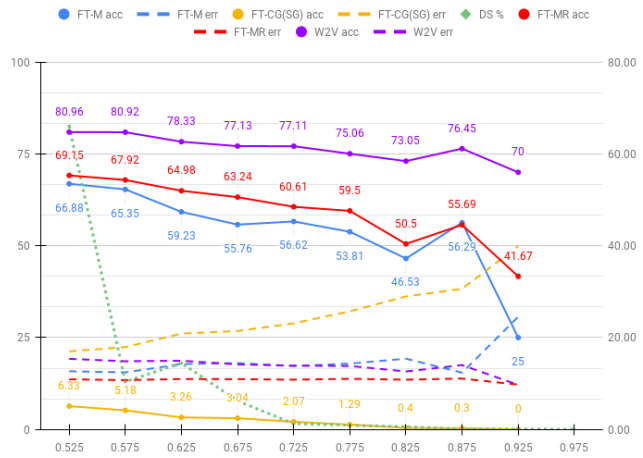

FIGURE 11. Relatedness-classification accuracies and errors as orthographic similarity of word-pairs increase from Q3 to Q4. Errors in dashes, accuracies in lines. Ran on English `editsim` dataset. x-axis orthographic similarity from (0.5 to 1), y-axis shows the percentage. FT-MR red, FT-M blue, FT-CG(SG) yellow. The percentage of word-pair instances plotted in green diamonds. The significant decrease after 85% orthographic-similarity is due data scarcity (green lines).

the data heavily skewed toward the related side. However, the evaluation focuses on the minority (unrelated) side (Table 19). 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 10. 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 VI-D5) is not noticeable when the orthographic similarity of word pairs occurs naturally.

b: 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

TABLE 21. Experiment 2b: Word relatedness experiments on combined WordSim datasets.

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

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.

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 19).

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 word-pairs 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 20). 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 Fig. 9 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 Fig. 10 and the Appendix for the distributions of all models).

4) SHIFTED CHAR-GRAM SPACE

As demonstrated in Table 18, 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. Fig. 10 also illustrates the distribution of the OSimUnr dataset, where all the ground-truth values are equally distributed (including

TABLE 22. Experiment 2a: Word relatedness experiments on Wordsim datasets. Language means are calculated by getting the average of relatedness datasets (similarity 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

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 20). It becomes evident that the decline in performance observed in the subword-level experiments can be attributed to the FT-CG segmentation.

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 18). 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 22) in word relatedness and approximately 1% in relatedness classification (Table 18).

B. 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 (Fig. 11). 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.

Fig. 11 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 Fig. 11. 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 Fig. 11 (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.

C. WORD RELATEDNESS

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 22). 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 [88], 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 22).

The benchmark LLMs achieved the highest scores in English (0.81, as shown in Table 21), 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.

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 23). 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 23 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) [6], BPE [7], and MorphMine [91], as well as supervised models like CHIPMUNK (sms) [9] and Spacy [93] with 'weighted + PC removal - LST'. Table 23 is an exhaustive list of studies that report results on the AnlamVer dataset and citing its paper.[31] We exclude the experiments reported by Tulu [92] 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. [89], 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 [94]) to the target language through automatic translation and post-processing (i.e., retrofitting) after semantic training.

TABLE 23. 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	
	FT-M (SG)	0.69	0.40	
	FT-MR	0.63	**0.44**	
[10]	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	
[89]	CLSRI-PS	0.61	-	ρ
	FT - Distributional	0.53	-	
[90]	weighted + PC removal - BPE	-	0.41	ρ_2
	weighted + PC removal - LST	-	0.29	
[91]	MorphMine	0.49	-	ρ
	Morfessor	0.48	-	
	BPE	0.47	-	
[92]	FT - SkipGram	0.80*	-	ρ
	Glove	0.77*	-	

3) VISUALIZATION

We present a t-SNE [95] visualization of the affix vector representations trained (based on the FT-M configuration) in this study (see Fig. 12). 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 Appendix.

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

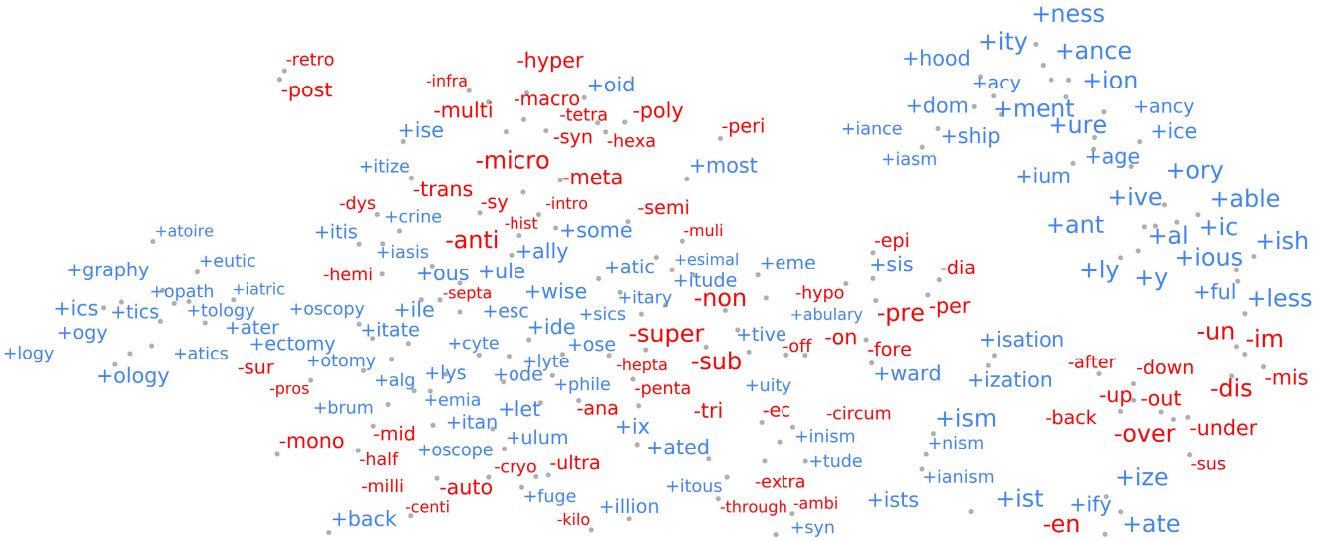

FIGURE 12. t-SNE visualization of affix vectors for English from the FT-M model configuration. Red for prefixes, blue for suffixes. Font size corresponds to frequency groups. Refer to Appendix for full plots.

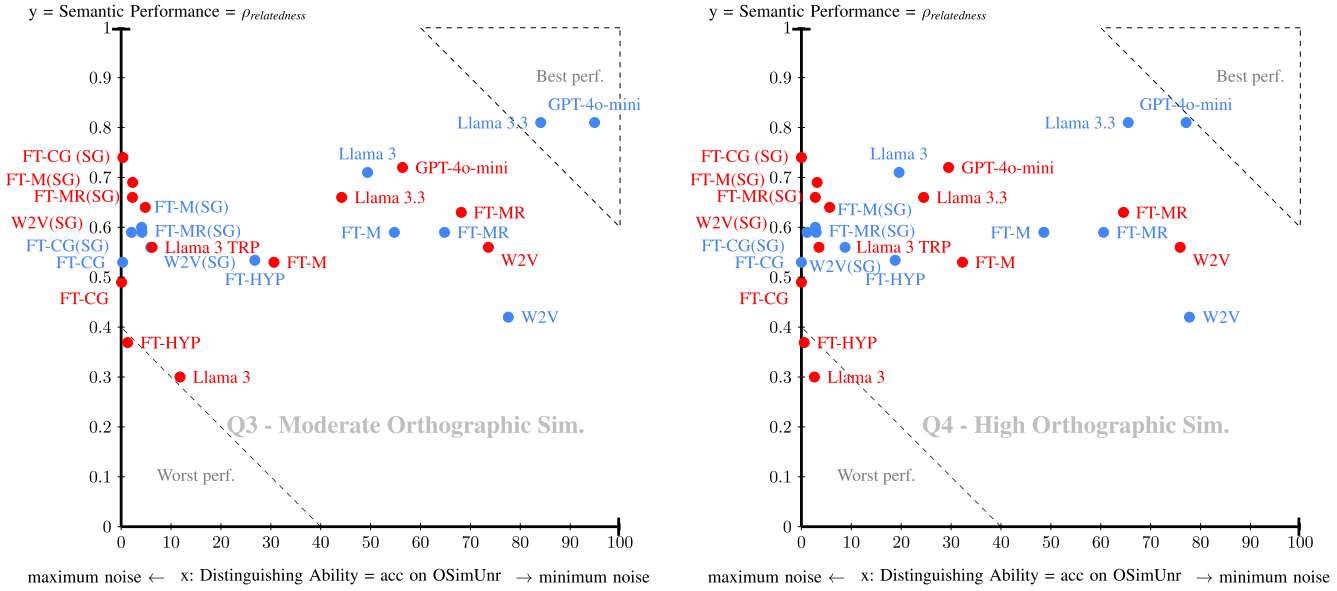

FIGURE 13. Semantic Clarity Space (Q3 on the left, Q4 on the right) - Illustrating semantic performance and distinguishing capabilities of various model configurations. Y axis: Relative semantic performance task: Spearman (ρ) scores of word relatedness on aggregate dataset (Table 22). X axis: Accuracy scores of unrelatedness-identification task OSimUnr (over_ft23) (Table 18). Turkish in red, English in blue dots. Only W2V is at the word-level. OOVs excluded from the word-level experiments.

TABLE 24. Semantic Clarity Index (SCI) scores of various model configurations (excluding LLM benchmarks).

	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

D. INTERPRETATIONS

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 20). It is 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

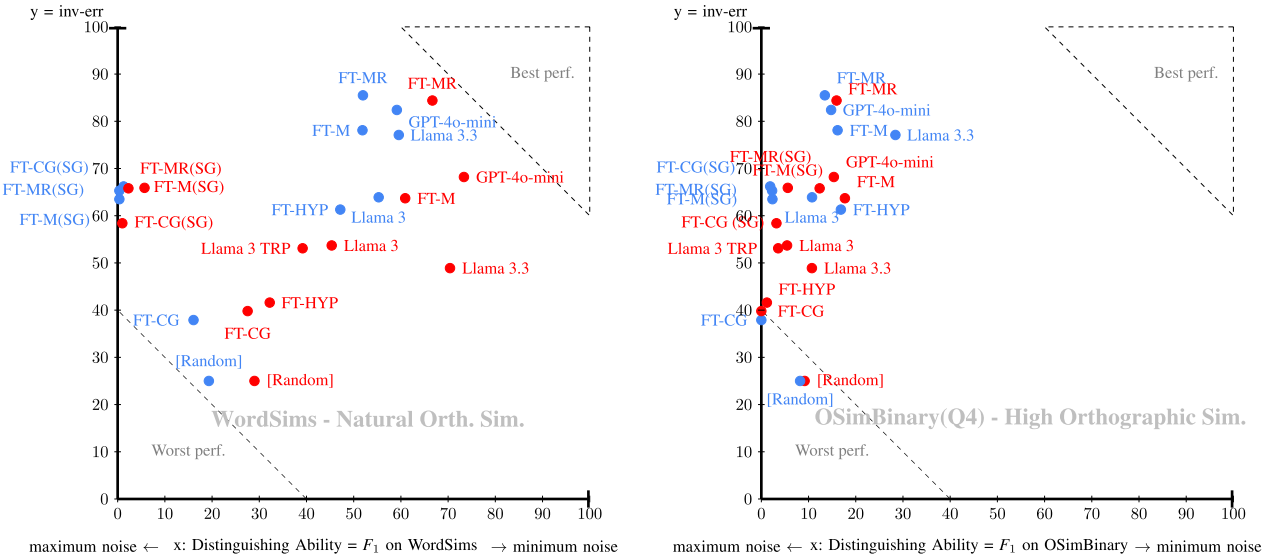

FIGURE 14. Alternative Semantic Clarity Space with Different Metrics (WordSims on the left, OSimBinary-Q4 on the right) Y axis: Inverted err = 100 − *err* scores of relatedness-classification task on the over_ft23 Q3 ds. (Table 18). X axis: F_1 scores of relatedness-classification task (Table 19). Turkish in red, English in blue dots.

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.

1) BAG-OF-AFFIX-MORPHEMES

Although there are some discernible clusters in t-SNE visualizations (Fig. 12), 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. [18], learning morphemes with different coefficient weights based on their types can be advantageous.

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 [96] 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.

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.

Figs. 11, 13, and 14 collectively illustrate that an increase in the generation of meaningless units through segmentation leads to what we refer to as 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.

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 Fig. 13, 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): $sci(sem, dist) = min(sem, dist)$. This metric encourages a balance between acquiring relationships between concepts and simultaneously discerning the distinctions among them. As depicted in Table 24, 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 (24) 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. 14 demonstrates an alternative space that utilizes continuous error from Experiment 1 and F_1 score for the OSimBinary dataset.

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.

6) ROLE OF MORPHOLOGY

In tasks that necessitate the handling of 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 [17].

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 foundation models (i.e., LLMs), particularly through advanced enrichment methods such as fine-tuning and retrieval-augmented generation.

VII. CONCLUSION

This study 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 tasks, offering insights for enhancing semantic space modeling in the realm of natural language processing.

APPENDIX A
TABLES AND FIGURES

See Figures 15–17 and Table 25.

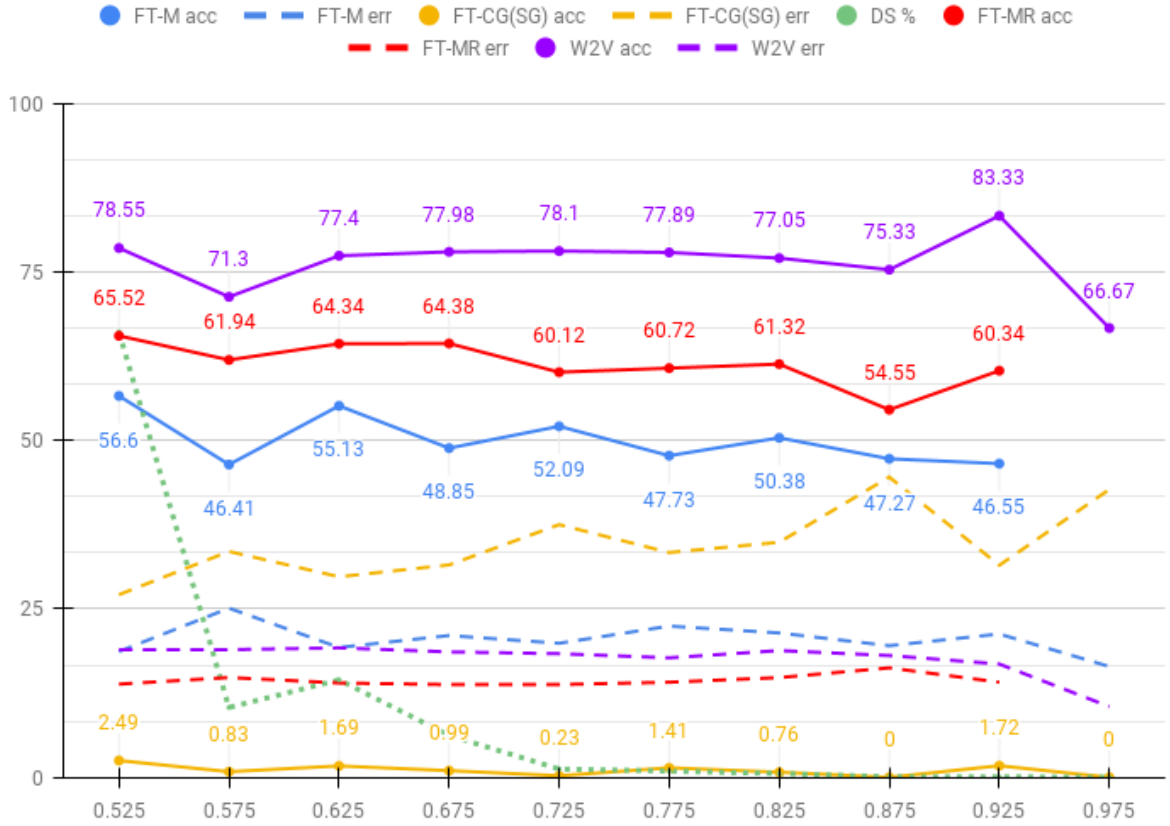

FIGURE 15. Relatedness-classification accuracies and errors as orthographic similarity of word-pairs increase from Q3 to Q4. Errors in dashes, accuracies in lines. x-axis orthographic similarity from (0.5 to 1), y-axis shows the percentage. The percentage of word-pair instances plotted in green diamonds.

TABLE 25. 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		-		-		0-1			0-1			0-max			0-1			0-max			0-max		
Info. Content		-		-		no			no			no			yes			yes			yes		
Measure	**OOV%**	**Acc.**	F_1	**Acc.**	F_1	**Acc.**	F_1	ρ	**Acc.**	F_1	ρ	**Acc.**	F_1	ρ	**Acc.**	F_1	ρ	**Acc.**	F_1	ρ	**Acc.**	F_1	ρ
	UNR%	**Pre.**	**Rec.**	**Pre.**	**Rec.**	Pre.	**Rec.**		**Pre.**	**Rec**		**Pre.**	**Rec.**		**Pre.**	**Rec.**		**Pre.**	**Rec.**		**Pre.**	**Rec.**	
RG	0	0.52	0.58	0.62	0.76	0.63	0.76	0.76	0.69	0.67	**0.78**	0.60	0.75	**0.78**	**0.74**	**0.77**	**0.78**	0.58	0.49	0.77	**0.74**	0.75	**0.78**
	38.46	0.64	0.53	0.62	**1.00**	0.64	0.93		**1.00**	0.50		0.61	**0.95**		0.85	0.70		**1.00**	0.33		0.90	0.65	
MC	0	0.53	0.63	0.60	0.75	0.63	0.76	0.75	0.77	0.76	0.72	0.63	0.76	0.72	0.77	0.80	0.75	0.63	0.56	**0.82**	**0.80**	**0.81**	0.73
	40	0.60	0.67	0.60	**1.00**	0.63			**1.00**	0.61		0.63	**0.94**		0.82	0.78		**1.00**	0.39		0.93	0.72	
WordSim353	1.42	0.47	0.62	**0.90**	**0.95**	0.81	0.89	**0.35**	0.30	0.36	0.31	**0.89**	**0.94**	0.31	0.60	0.74	0.31	0.14	0.08	0.30	0.53	0.67	**0.35**
	9.77	0.88	0.48	0.90	**1.00**	0.90	0.89		0.99	0.22		0.90	**0.98**		0.92	0.61		**1.00**	0.04		0.92	0.52	
RareWords	55.26	0.50	0.64	**0.88**	**0.94**	0.86	0.92	0.24	0.78	0.87	**0.28**	**0.87**	**0.93**	0.28	0.63	0.75	0.21	0.18	0.12	0.18	0.81	0.89	**0.31**
	12.97	0.88	0.50	0.88	**1.00**	0.88	0.97		0.90	0.84		0.87	**0.99**		0.89	0.66		**0.98**	0.06		0.91	0.87	
MEN	11.43	0.51	0.62	**0.79**	**0.88**	0.74	0.85	0.39	0.35	0.29	0.39	0.78	**0.87**	0.39	0.59	0.67	0.36	0.23	0.04	0.37	0.54	0.60	**0.40**
	21.83	0.80	0.50	0.79	**1.00**	0.80	0.90		0.99	0.17		0.79	0.98		0.90	0.54		**1.00**	0.02		0.93	0.44	
MTurk771	0	0.50	0.66	**0.95**	0.97	**0.95**	**0.98**	0.45	0.81	0.89	**0.49**	**0.95**	0.97	**0.49**	**0.95**	0.97	**0.49**	0.19	0.26	0.48	0.92	0.96	0.40
	5.06	0.96	0.50	0.95	**1.00**	0.95	1.00		**0.98**	0.81		0.95	**1.00**		0.97	0.98		0.97	0.15		0.97	0.94	
EN Rel.	23.54	0.50	0.62	**0.82**	**0.90**	0.78	0.87	0.35	0.50	0.58	0.35	**0.80**	**0.89**	0.35	0.63	0.74	0.29	0.23	0.10	0.28	0.63	0.73	**0.38**
	16.90	0.82	0.50	0.82	**1.00**	0.82	0.93		0.91	0.43		0.81	**0.98**		0.87	0.64		**0.93**	0.05		0.90	0.61	
AnlamVer-Rel	26.20	0.50	0.57	0.67	0.80	**0.72**	**0.83**	**0.36**	0.48	0.49	0.28	0.70	0.82	0.28	-	-	-	-	-	-	-	-	-
	30.62	0.67	0.49	0.67	**1.00**	0.72	**0.97**		**0.77**	0.36		0.71	**0.97**		-	-		-	-		-	-	
Sopaoglu	2.97	0.50	0.56	0.62	0.77	0.67	**0.79**	0.65	**0.74**	0.75	**0.71**	0.66	**0.79**	0.70	-	-	-	-	-	-	-	-	-
	36.73	0.62	0.51	0.62	**1.00**	0.67	0.97		**1.00**	0.60		0.66	**0.98**		-	-		-	-		-	-	
TR Rel.	22.64	0.50	0.57	0.66	0.50	**0.71**	**0.82**	**0.41**	0.53	0.54	0.36	0.69	0.81	0.36	-	-	-	-	-	-	-	-	-
	32.10	0.66	0.50	0.66	**1.00**	0.71	**0.97**		**0.82**	0.40		0.70	**0.97**		-	-		-	-		-	-	

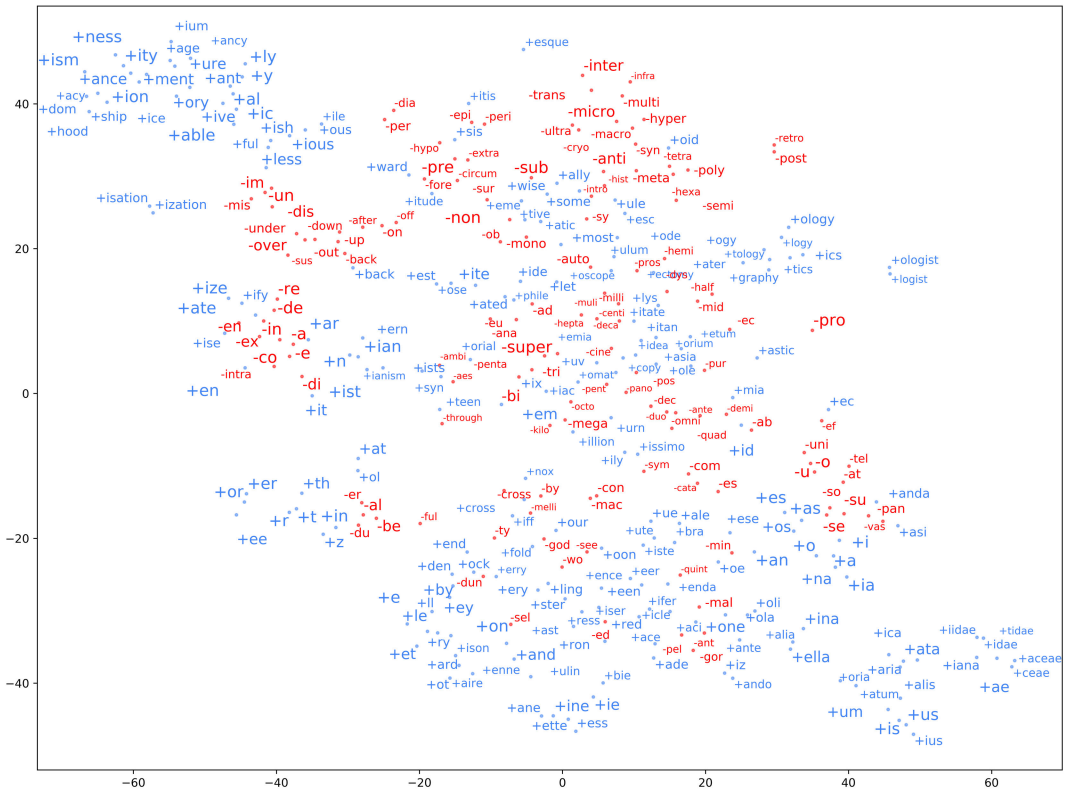

FIGURE 16. t-SNE visualization of affix vectors for English from the FT-M model configuration. Prefixes in red, suffixes in blue. More frequent affixes are displayed with bigger fonts. The affixes less frequent than 50 are removed.

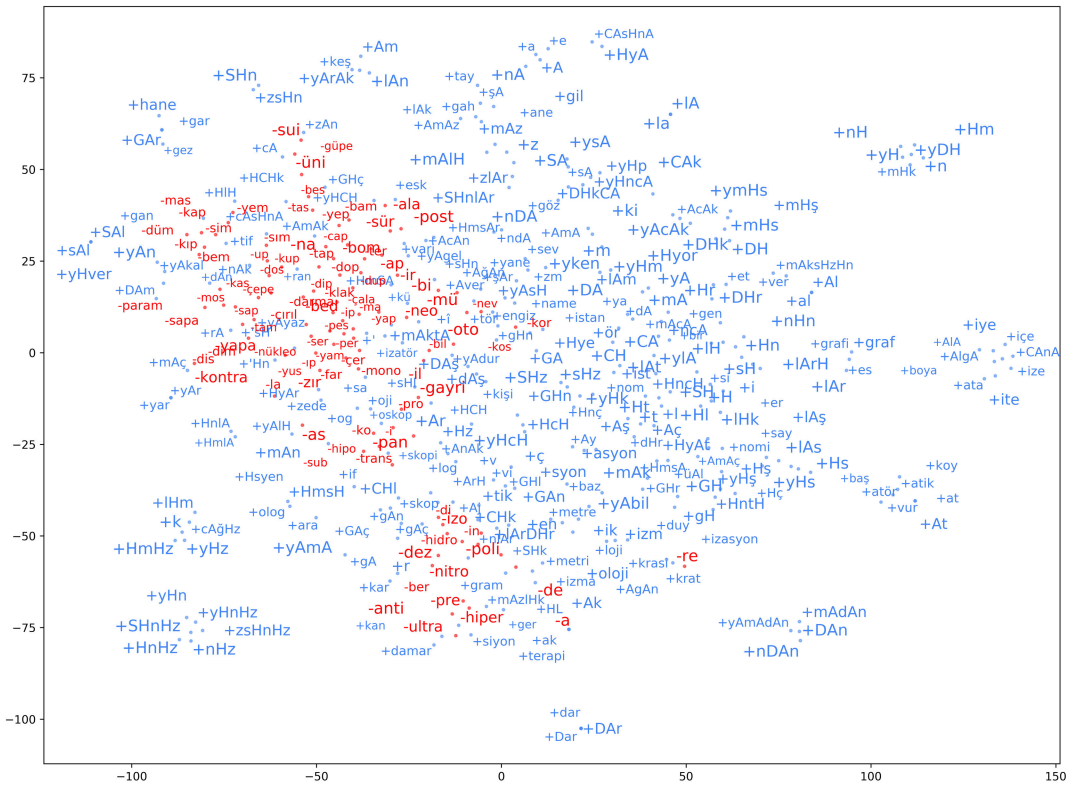

FIGURE 17. t-SNE visualization of affix vectors for Turkish from the FT-M model configuration. Prefixes in red, suffixes in blue. More frequent affixes are displayed with bigger fonts. The affixes less frequent than 50 are removed.

APPENDIX B

See Figure 18.

APPENDIX C

See Figure 19.

APPENDIX D

See Figure 20.

APPENDIX E

See Figure 21.

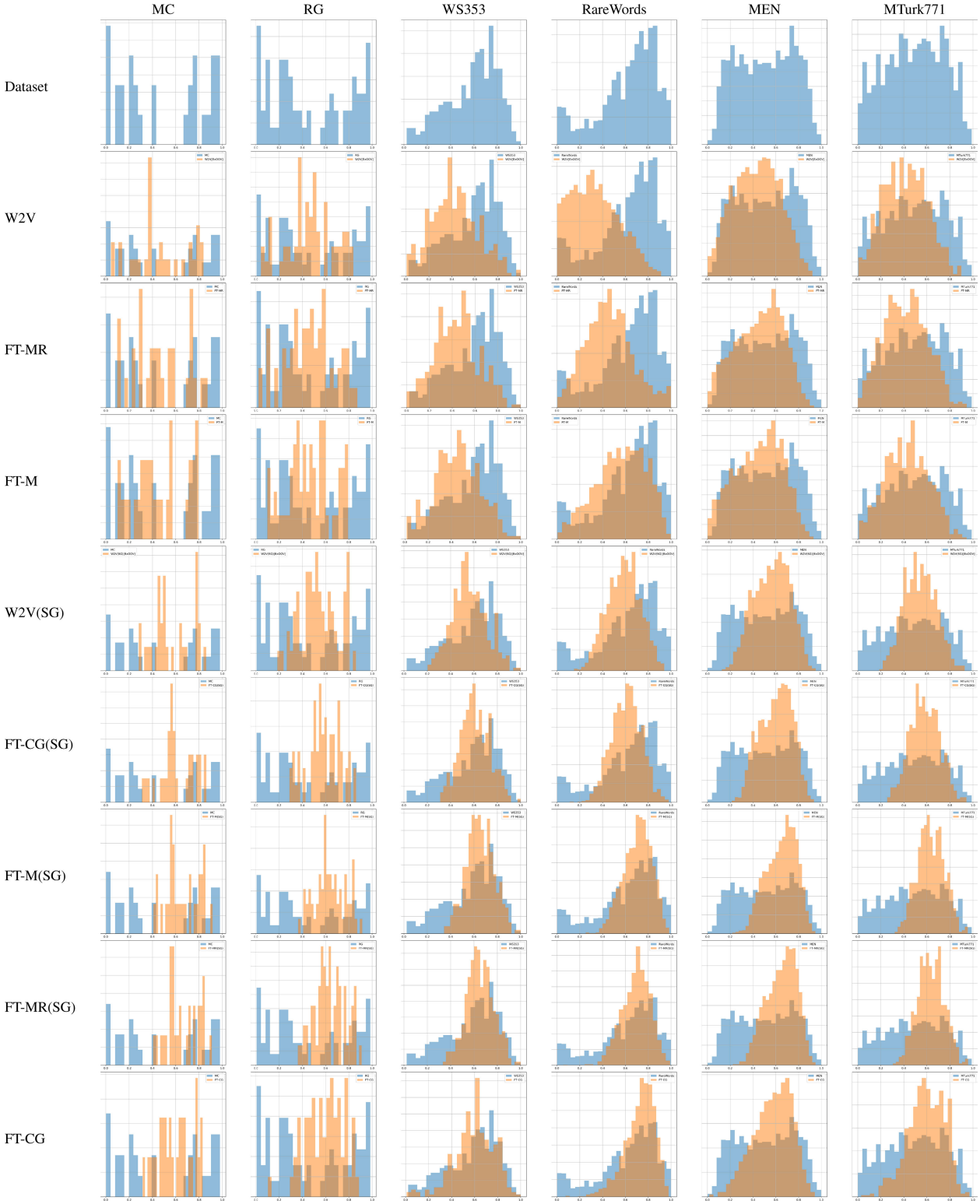

FIGURE 18. English -Model distributions on relatedness datasets.

FIGURE 19. Turkish - Model Distributions on Relatedness and Similarity Datasets

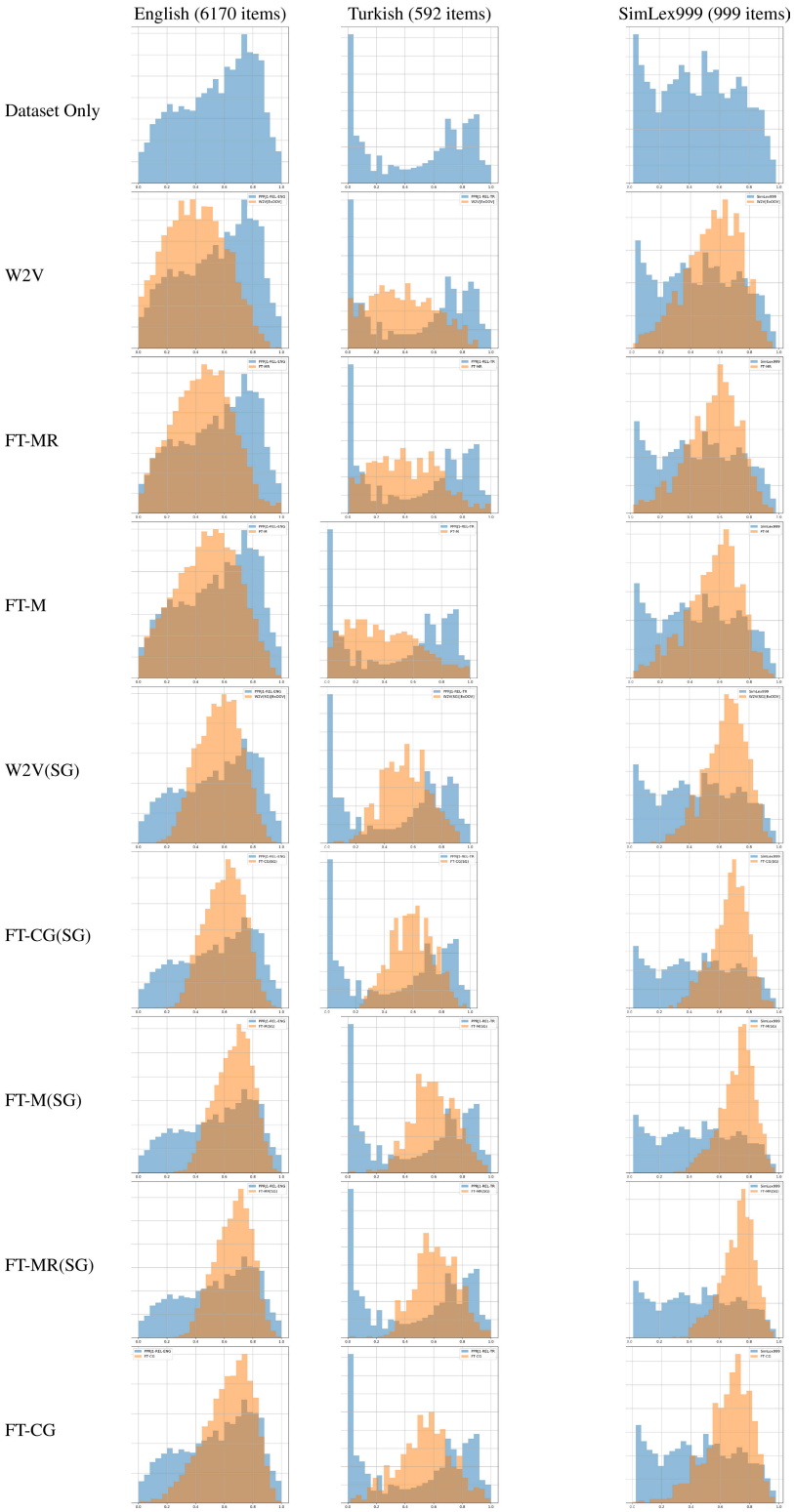

FIGURE 20. Model Distributions on Aggregate Relatedness and SimLex999 Datasets.

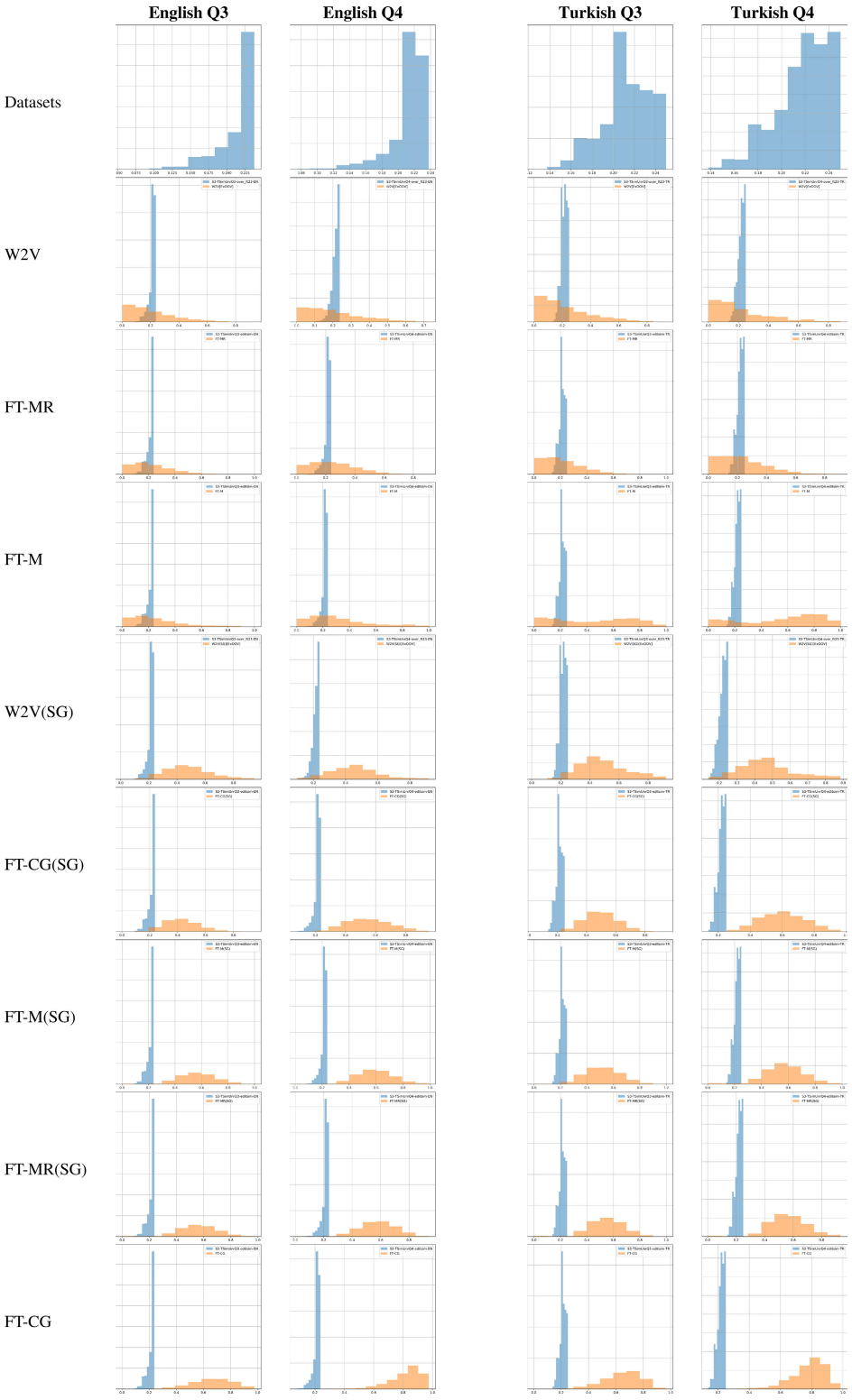

FIGURE 21. English - Model Distributions on OSimUnr Dataset (`editsim` and `over_ft23` mixed).

TABLE 26. 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

APPENDIX F

See Table 26.

REFERENCES

[1] C. H. Sánchez-Gutiérrez, H. Mailhot, S. H. Deacon, and M. A. Wilson, "MorphoLex: A derivational morphological database for 70,000 English words," *Behav. Res. Methods*, vol. 50, no. 4, pp. 1568–1580, Aug. 2018.

[2] G. K. Zipf, *The Psychobiology of Language*. New York, NY, USA: Houghton-Mifflin, 1935.

[3] P. W. Anderson, "More is different: Broken symmetry and the nature of the hierarchical structure of science," *Science*, vol. 177, no. 4047, pp. 393–396, Aug. 1972.

[4] M. Baroni, "Linguistic generalization and compositionality in modern artificial neural networks," 2019, *arXiv:1904.00157*.

[5] A. Göksel and C. Kerslake, *Turkish: A Comprehensive Grammar*. Evanston, IL, USA: Routledge, 2004.

[6] S. Virpioja, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," Aalto Univ., Espoo, Finland, Tech. Rep., 2013.

[7] P. Gage, "A new algorithm for data compression," *C Users J.*, vol. 12, no. 2, pp. 23–38, Feb. 1994.

[8] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018, *arXiv:1808.06226*.

[9] R. Cotterell, T. Müller, A. Fraser, and H. Schütze, "Labeled morphological segmentation with semi-Markov models," in *Proc. 19th Conf. Comput. Natural Lang. Learn.*, 2015, pp. 164–174.

[10] Y. Zhu, I. Vulić, and A. Korhonen, "A systematic study of leveraging subword information for learning word representations," in *Proc. Conf. North*, 2019, pp. 912–932.

[11] J. Ryland Williams, P. R. Lessard, S. Desu, E. M. Clark, J. P. Bagrow, C. M. Danforth, and P. Sheridan Dodds, "Zipf's law holds for phrases, not words," *Sci. Rep.*, vol. 5, no. 1, pp. 1–7, Aug. 2015.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 3111–3119.

[14] É. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proc. 11th Int. Conf. Language Resour. Eval. (LREC)*, Jan. 2018.

[15] B. Heinzerling and M. Strube, "BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*. Miyazaki, Japan: ELRA, 2018.

[16] E. M. Bender, "Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax," *Synth. Lectures Hum. Lang. Technol.*, vol. 6, no. 3, pp. 1–184, Jun. 2013.

[17] R. Sutton. (2019). *The Bitter Lesson*. Accessed: Oct. 17, 2019. [Online]. Available: http://www.incompleteideas.net/IncIdeas/BitterLesson.html

[18] S. Qiu et al., "Co-learning of word representations and morpheme does play," in *Proc. 25th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2014, pp. 141–150.

[19] J. Zhao, S. Mudgal, and Y. Liang, "Generalizing word embeddings using bag of subwords," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 601–606.

[20] V. Romanov and A. Khusainova, "Evaluation of morphological embeddings for the Russian language," in *Proc. 3rd Int. Conf. Natural Lang. Process. Inf. Retr.* New York, NY, USA: ACM, Jun. 2019, pp. 144–148, doi: 10.1145/3342827.3342846.

[21] G. Ercan and O. T. Yıldız, "AnlamVer: Semantic model evaluation dataset for Turkish-word similarity and relatedness," in *Proc. 27th Int. Conf. Comput. Linguistics*, Aug. 2018, pp. 3819–3836.

[22] A. Üstün, M. Kurfalı, and B. Can, "Characters or morphemes: How to represent words?" in *Proc. 3rd Workshop Represent. Learn.*, 2018, pp. 144–153.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," 2018, *arXiv:1804.07461*.

[24] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," 2020, *arXiv:2009.03300*.

[25] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei, "Challenging BIG-bench tasks and whether chain-of-thought can solve them," 2022, *arXiv:2210.09261*.

[26] E. M. Bender and A. Koller, "Climbing towards NLU: On meaning, form, and understanding in the age of data," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5185–5198.

[27] G. Ercan. (2024). *OSimUnr*. [Online]. Available: https://github.com/gokhanercan/OSimUnr

[28] Z. S. Harris, "Distributional structure," *Word*, vol. 10, nos. 2–3, pp. 146–162, Aug. 1954.

[29] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating semantic models with (Genuine) similarity estimation," *Comput. Linguistics*, vol. 41, no. 4, pp. 665–695, Dec. 2016.

[30] F. De Saussure, W. Baskin, and P. Meisel, *Course in General Linguistics*. New York, NY, USA: Columbia Univ. Press, 2011.

[31] S. Hengchen and N. Tahmasebi, "SuperSim: A test set for word similarity and relatedness in Swedish," 2021, *arXiv:2104.05228*.

[32] U. Salaev, E. Kuriyozov, and C. Gómez-Rodríguez, "SimRelUz: Similarity and relatedness scores as a semantic evaluation dataset for uzbek language," 2022, *arXiv:2205.06072*.

[33] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Trans. Assoc. Comput. Linguistics*, vol. 3, pp. 211–225, Dec. 2015.

[34] T. Kliegr and O. Zamazal, "Antonyms are similar: Towards paradigmatic association approach to rating similarity in SimLex-999 and WordSim-353," *Data & Knowl. Eng.*, vol. 115, pp. 174–193, Apr. 2018.

[35] I. Vulić, S. Baker, E. M. Ponti, U. Petti, I. Leviant, K. Wing, O. Majewska, E. Bar, M. Malone, T. Poibeau, R. Reichart, and A. Korhonen, "Multi-SimLex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity," *Comput. Linguistics*, vol. 46, no. 4, pp. 847–897, Feb. 2021, doi: 10.1162/coli_a_00391.

[36] G. Lapesa, S. Evert, and S. S. Walde, "Contrasting syntagmatic and paradigmatic relations: Insights from distributional semantic models," in *Proc. 3rd Joint Conf. Lexical Comput. Semantics (SEM)*, 2014, pp. 160–170.

[37] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Doklady*, vol. 163, no. 4, pp. 845–848, Jan. 1965.

[38] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," in *Proc. 10th Int. Conf. World Wide Web*, vol. 20, Jan. 2002, pp. 406–414.

[39] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun. ACM*, vol. 8, no. 10, pp. 627–633, Oct. 1965.

[40] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Lang. Cognit. Processes*, vol. 6, no. 1, pp. 1–28, Jan. 1991.

[41] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, May 2013, pp. 746–751.

[42] A. Gladkova and A. Drozd, "Intrinsic evaluations of word embeddings: What can we do better?" in *Proc. 1st Workshop Evaluating Vector-Space Represent. (NLP)*, 2016, pp. 36–42.

[43] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, "Problems with evaluation of word embeddings using word similarity tasks," 2016, *arXiv:1605.02276*.

[44] M. A. Hadj Taieb, T. Zesch, and M. Ben Aouicha, "A survey of semantic relatedness evaluation datasets and procedures," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4407–4448, Aug. 2020.

[45] C. Spearman, "The proof and measurement of association between two things," Appleton-Century-Crofts, Tech. Rep., 1961.

[46] E. Bruni, N. K. Tran, and M. Baroni, "Multimodal distributional semantics," *J. Artif. Intell. Res.*, vol. 49, pp. 1–47, Jan. 2014.

[47] T. Zesch and I. Gurevych, "Automatically creating datasets for measures of semantic relatedness," in *Proc. Workshop Linguistic Distances (LD)*, 2006, pp. 16–24.

[48] T. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology," in *Proc. CoNLL*, Aug. 2013, pp. 104–113.

[49] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 1406–1414.

[50] U. Sopaoğlu and G. Ercan, "Evaluation of semantic relatedness measures for Turkish language," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*, Jan. 2018, pp. 600–611.

[51] F. Karlsson, *Yleinen Kielitiede*. Helsinki Univ. Press, 1998.

[52] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Commun. ACM*, vol. 18, no. 6, pp. 341–343, Jun. 1975.

[53] G. Kondrak, "N-gram similarity and distance," in *Proc. Int. Symp. String Process. Inf. Retr.* Cham, Switzerland: Springer, Jan. 2005, pp. 115–126.

[54] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The best of both worlds," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 31–39, Mar. 2011.

[55] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[56] R. Al-Rfou, B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual NLP," in *Proc. 17th Conf. Comput. Natural Lang. Learn.* Sofia, Bulgaria: ACM, Aug. 2013, pp. 183–192. [Online]. Available: https://www.aclweb.org/anthology/W13-3520

[57] K. Koskenniemi, "Two-level morphology: A general computational model for word-form recognition and production," Dept. General Linguistics Helsinki, Univ. Helsinki, Helsinki, Finland, Tech. Rep., 1983, vol. 11.

[58] O. T. Yıldız, B. Avar, and G. Ercan, "An open, extendible, and fast Turkish morphological analyzer," in *Proc. Natural Lang. Process. Deep Learn. World*, Varna, Bulgaria, Oct. 2019, pp. 1364–1372. [Online]. Available: https://www.aclweb.org/anthology/R19-1156

[59] B. N. Arıcan et al., "Morpholex Turkish: A morphological lexicon for Turkish," in *Proc. 13th Lang. Resour. Eval. Conf. Globalex Workshop Linked Lexicography*, 2022, pp. 68–74.

[60] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[61] Ö. Bakay et al., "Turkish WordNet KeNet," in *Proc. 11th Global Wordnet Conf.*, Jan. 2021, pp. 166–174.

[62] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet: Similarity: Measuring the relatedness of concepts," in *Proc. Demonstration Papers HLT-NAACL XX (HLT-NAACL)*, 2004, pp. 38–41.

[63] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 133–138.

[64] C. Leacock, "Combining local context and WordNet similarity for word sense identification," *WordNet, Electron. Lexical Database*, vol. 49, no. 2, pp. 265–283, 1998.

[65] D. Lin, "An information-theoretic definition of similarity," in *Proc. ICML*. Citeseer, Jul. 1998, pp. 296–304.

[66] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," 1997, *arXiv:cmp-lg/9709008*.

[67] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," 1995, *arXiv:cmp-lg/9511007*.

[68] A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of lexical semantic relatedness," *Comput. Linguistics*, vol. 32, no. 1, pp. 13–47, Mar. 2006.

[69] Z. Zhang, A. L. Gentile, and F. Ciravegna, "Recent advances in methods of lexical semantic relatedness – a survey," *Natural Lang. Eng.*, vol. 19, no. 4, pp. 411–479, Oct. 2013.
[70] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet, Electron. Lexical Database*, vol. 1998, pp. 305–332, May 1998.
[71] R. E. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1.
[72] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, 2009, pp. 19–27.
[73] R. Ehsani, E. Solak, and O. T. Yildiz, "Constructing a WordNet for Turkish using manual and automatic annotation," *ACM Trans. Asian Low-Resource Language Inf. Process.*, vol. 17, no. 3, pp. 1–15, Sep. 2018.
[74] H. Mailhot, M. A. Wilson, J. Macoir, S. H. Deacon, and C. Sánchez-Gutiérrez, "MorphoLex-FR: A derivational morphological database for 38,840 French words," *Behav. Res. Methods*, vol. 52, no. 3, pp. 1008–1025, Jun. 2020.
[75] F. Bond, L. M. D. Costa, M. W. Goodman, J. P. McCrae, and A. Lohk, "Some issues with building a multilingual wordnet," in *Proc. 12th Lang. Resour. Eval. Conf.*, May 2020, pp. 3189–3197.
[76] K. Batsuren et al., "UniMorph 4.0: Universal morphology," 2022, *arXiv:2205.03608*.
[77] Z. Žabokrtskỳ et al., "Towards universal segmentations: Unisegments 1.0," in *Proc. 13th Lang. Resour. Eval. Conf.*, 2022, pp. 1137–1149.
[78] K. Batsuren, G. Bella, and F. Giunchiglia, "MorphyNet: A large multilingual database of derivational and inflectional morphology," in *Proc. 18th SIGMORPHON Workshop Comput. Res. Phonetics, Phonology, Morphology*, 2021, pp. 39–48.
[79] L. von Ahn, "Games with a purpose," *Comput.*, vol. 39, no. 6, pp. 92–94, Jun. 2006.
[80] A. Lazaridou, M. Marelli, R. Zamparelli, and M. Baroni, "Compositionally derived representations of morphologically complex words in distributional semantics," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2013, pp. 1517–1526.
[81] H. Sak, T. Güngör, and M. Saraçlar, "Resources for Turkish morphological processing," *Lang. Resour. Eval.*, vol. 45, no. 2, pp. 249–261, May 2011.
[82] P. Lison, J. Tiedemann, and M. Kouylekov, "Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018.
[83] A. Safaya, E. Kurtuluş, A. Göktoğan, and D. Yuret, "Mukayese: Turkish NLP strikes back," 2022, *arXiv:2203.01215*.
[84] J. Bhattacharjee, *FastText Quick Start Guide: Get Started With Facebook's Library for Text Representation and Classification*. Packt, 2018.
[85] A. Grattafiori et al., "The llama 3 herd of models," 2024, *arXiv:2407.21783*.
[86] A. Hurst et al., "GPT-4o system card," 2024, *arXiv:2410.21276*.
[87] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.
[88] O. İrsoy, A. Benton, and K. Stratos, "Corrected CBOW performs as well as skip-gram," 2020, *arXiv:2012.15332*.
[89] E. M. Ponti, I. Vulić, G. Glavaš, R. Reichart, and A. Korhonen, "Cross-lingual semantic specialization via lexical relation induction," in *Proc. 9th Int. Joint Conf. Natural Lang. Process. Conf. Empirical Methods Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 2206–2217.
[90] D. Bollegala, R. Kiryo, K. Tsujino, and H. Yukawa, "Language-independent tokenisation rivals language-specific tokenisation for word similarity prediction," 2020, *arXiv:2002.11004*.
[91] A. El-Kishky, F. Xu, A. Zhang, and J. Han, "Parsimonious morpheme segmentation with an application to enriching word embeddings," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 64–73.
[92] C. Tulu, "Experimental comparison of pre-trained word embedding vectors of Word2Vec, glove, fasttext for word level semantic text similarity measurement in Turkish," *Adv. Sci. Technol. Res. J.*, vol. 16, no. 4, pp. 147–156, 2022.
[93] M. Honnibal and I. Montani, "SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," Tech. Rep., 2017.
[94] B. Kipfer, *Roget's 21st Century Thesaurus*, 3rd ed., New York, NY, USA: Philip Lief, 2005.
[95] L. Van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, Jan. 2008.
[96] M. Baroni and R. Zamparelli, "Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2010, pp. 1183–1193.

GÖKHAN ERCAN received the master's degree in software engineering from Boğaziçi University, Istanbul, Türkiye. He is currently an Engineering Leader at BlueCloud. His research interests include natural language processing, information retrieval, software architecture, and AI for code. In 2018, he received the Best Resource Paper Award from the COLING Conference held in Santa Fe, NM, USA.

OLCAY TANER YILDIZ received his PhD degree in computer science from Boğaziçi University, Istanbul, Turkey and did postdoctoral work at the University of Minnesota, in 2005. He was with the Department of Computer Engineering at Işık University, between 2005-2020. He is a full professor in Özyeğin University since 2020. His research interests include natural language processing, machine learning, and bioinformatics.