Complementary Strategies for Low Resourced Morphological Modeling

Alexander Erdmann and Nizar Habash
Computational Approaches to Modeling Language Lab
New York University Abu Dhabi
United Arab Emirates
{ae1541,nizar.habash}@nyu.edu

Abstract

Morphologically rich languages are challenging for natural language processing tasks due to data sparsity. This can be addressed either by introducing out-of-context morphological knowledge, or by developing machine learning architectures that specifically target data sparsity and/or morphological information. We find these approaches to complement each other in a morphological paradigm modeling task in Modern Standard Arabic, which, in addition to being morphologically complex, features ubiquitous ambiguity, exacerbating sparsity with noise. Given a small number of out-of-context rules describing closed class morphology, we combine them with word embeddings leveraging subword strings and noise reduction techniques. The combination outperforms both approaches individually by about 20% absolute. While morphological resources already exist for Modern Standard Arabic, our results inform how comparable resources might be constructed for non-standard dialects or any morphologically rich, low resourced language, given scarcity of time and funding.

1 Introduction

Morphologically rich languages pose many challenges for natural language processing tasks. This often takes the shape of data sparsity, as the increase in the number of possible inflections for any given core concept leads to a lower average word frequency of individual (i.e., unique) word types. Hence, models have fewer chances to learn about types based on their in-context behavior. One common, albeit time consuming response to this challenge is to introduce out-of-context morphological knowledge, hand crafting rules to relate forms inflected from the same lemma. The other common response is to adopt machine learning architectures specifically targeting data sparsity and/or morphological information. We find these two responses to be complementary in a paradigm modeling task for Modern Standard Arabic (MSA).

MSA is characterized by morphological richness and extreme orthographic ambiguity, compounding the issue of data sparsity with noise (Habash, 2010). Despite its challenges, MSA is relatively well resourced, with many solutions for morphological analysis and disambiguation leveraging large amounts of annotated data, hand crafted rules, and/or sophisticated neural architectures (Khoja and Garside, 1999; Habash and Rambow, 2006; Smrž, 2007; Graff et al., 2009; Pasha et al., 2014; Abdelali et al., 2016; Inoue et al., 2017; Zalmout and Habash, 2017). Such resources and techniques, however, are not available or not viable for the many under resourced and often mutually unintelligible dialects of Arabic (DA), which are similarly morphologically rich and highly ambiguous (Chiang et al., 2006; Erdmann et al., 2017). Many recent efforts seek to develop morphological resources for DA, but most are under developed or specific to one dialect (Habash et al., 2012; Eskander et al., 2013; Jarrar et al., 2014; Al-Shargi et al., 2016; Eskander et al., 2016a; Khalifa et al., 2016, 2017; Zribi et al., 2017; Zalmout et al., 2018; Khalifa et al., 2018).

This work does not aim to develop a full morphological analysis and disambiguation resource, but to inform how one might be most efficiently developed for any DA variety or similarly low resourced language, given scarcity of time and funding. For such a resource to be practical and easily extendable to new DA varieties, it must take as input the natural, highly ambiguous orthography. Thus, we do not rely on constructed phonological representations to clarify ambiguities, as is common practice when modeling morphology for its own sake (Cotterell et al., 2016, 2017). To in-

Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 54–65
Brussels, Belgium, October 31, 2018.
https://doi.org/10.18653/v1/P17

form how such a resource should be developed, we evaluate minimally rule based and unsupervised techniques for clustering words that belong to the same paradigm in MSA. We primarily use pre-existing MSA resources only for evaluation, constraining resource availability to emulate DA settings during training, as we lack the resources to evaluate our techniques in DA. Our best system combines a minimal set of rules describing closed class morphology with word embeddings that leverage subword strings and noise reduction strategies. The former, despite being cheaper and easier to produce than other rule-based systems, provides valuable out-of-context morphological knowledge, which the latter complements by modeling the in-context behavior of words and morphemes. Combining the techniques outperforms either individually by about 20% absolute.

2 Morphology and Ambiguity

Arabic morphology is structurally and functionally complex. Structurally, paradigms are relatively large. Component cells convey morpho-syntactic properties at a much finer granularity than English. Functionally, many morphological processes are non-concatenative, or *templatic*. Arabic roots are lists of de-lexicalized radicals, which must be mapped onto a template to derive a word. The derived word will then exhibit some predictable semantic and morpho-syntactic relationship to the root, based on its template. For example, the root ر د د *r d d*,[1] having to do with *responding*, could take a singular nominal template where geminates are collapsed, becoming رد *rd*, 'response', or a so-called *broken plural* template, separating the geminate with a long vowel to become ردود *rdwd*, 'responses'. Arabic orthography complicates the issue further, as diacritics marking short vowels, gemination, and case endings are typically not written. In addition to causing frequent lexical ambiguity among forms that are pronounced differently, this also causes templatic processes to appear to be concatenative or completely disappear. For example, deriving 'to cool' برد *brd* (*fully diacritized*, بَرَّد *bar∼ad*) from 'coldness' برد *brd* (*fully diacritized*, بَرد *bar.d*) involves doubling the second root letter and adding a short vowel before the third, yet these templatic changes usually disappear in the orthography.

[1] Arabic transliteration is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

Most templatic processes are derivational, deriving new core meanings with separate paradigms from a shared root. Inflectional processes generally concatenate affixes to a shared stem to realize different cells in the same paradigm. Broken plurals however, like ردود *rdwd*, are a notable exception, resulting from a templatic inflectional process. Approximately 55% of all plurals are broken (Alkuhlani and Habash, 2011).

Arabic is further characterized by frequent cliticization of prepositions, conjunctions, and object pronouns. Thus, a single syntactic word can take many cliticized forms, potentially becoming homonymous with inflections of unrelated lemmas or distinct cells in the same paradigm. The برد *brd*, 'response'–'coldness' ambiguity exemplifies this. The 'response' meaning interprets ب *b* as a cliticized preposition meaning 'with', while the 'coldness' meaning interprets ب *b* as the first root radical. To investigate how these morphological traits affect our ability to model paradigms, we define the following morphological structures.

Paradigm All words that share a certain lemma comprise a paradigm, e.g., in Figure 1, the paradigm of verbal lemma رَدّ *rad∼*, 'to respond', contains the four words connected to it by a solid line. Ambiguity within the paradigm is referred to as *syncretism*, and is very common in Arabic. For example, the present tense second person masculine singular form is syncretic with the third person feminine singular in verbs, as shown by ترد *trd*, 'you[masc.sing]/she respond(s)'. Additionally, orthography normalizes short vowel distinctions between past tense second person masculine, second person feminine, and first person forms (and sometimes third person feminine), thus causing رَدَدتَ *radadta*, رَدَدتِ *radadti*, and رَدَدتُ *radadtu*, respectively, to be orthographically syncretic. Cliticized forms can also cause unique syncretisms, e.g., بردنا *brdnA* has two possible interpretations from the same lemma بَرَّد *bar∼ad*, 'to cool'. If the final suffix نا *nA* is interpreted as a past tense verbal exponent, it means 'we cooled', whereas if it is interpreted as a cliticized personal pronoun, it becomes 'he/it cooled us'.

Subparadigm At or below the paradigm level, subparadigms are comprised of all words that share the same *lemma ambiguity*. Lemma ambiguity refers to the set of all lemmas a word could have been derived from out of context. Hence, برد

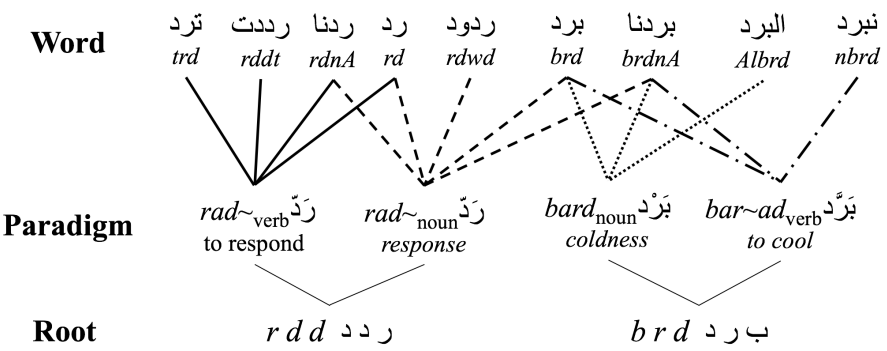

Figure 1: A clan of two families with two paradigms each, connected by both derivational and coincidental ambiguities. Line dotting style is only used to visually distinguish paradigm membership.

brd and بردنا *brdnA* form a subparadigm, being the only words in Figure 1 which can all be derived exclusively from lemmas, رَدّ *rad~*, 'response', بَرْد *bard*, 'coldness', and بَرَّد *bar~ad*, 'to cool'.

Family At or above the paradigm level, a family is comprised of all paradigms which can be linked via *derivational ambiguity*, such that all paradigms are derived from the same root. Thus, all forms mapping to the two paradigms which in turn map to the root ب ر د *b r d*, relating to *cold*, constitute a single family. The subparadigm of برد *brd* and بردنا *brdnA* link the two component paradigms via derivational ambiguity.[2]

Clan At or above the family level, a clan is comprised of all families which can be linked by *coincidental ambiguity*. Thus, the subparadigm of برد *brd* and بردنا *brdnA*, whose derivational ambiguity joins the paradigms of the ب ر د *b r d* family, also connects that family to the unrelated ر د د *r d d* family via coincidental ambiguity. This is caused by the multiple possible analyses of ب *b* as either a cliticized preposition or a root letter.

3 Experiments

In this section, we describe the data, design, and models used in our experiments.

[2]The linguistic concept of derivational family differs from ours in that it does not require any ambiguous forms to be shared by derivationally related paradigms. However, identifying such derivational families automatically is non-trivial. Even if the shared root can be identified, it can be difficult to determine whether the root is mono or polysemous, e.g., ر ع ش *š ʿ r* could refer to *hair*, *poetry*, or *feeling*. Regardless, our definition of family better serves our investigation into the effects of ambiguity.

3.1 Data

To train word embedding models, we use a corpus of 500,000 Arabic sentences (13 million words) randomly selected from the corpus used in Almahairi et al. (2016). This makes our findings more generalizable to DA, as many dialects have similar amounts of available data (Erdmann et al., 2018). We clean our corpus via standard preprocessing[3] and analyze each word out of context with SAMA (Graff et al., 2009) to get the set of possible fully diacritized lemmas from which it could be derived.[4]

To build an evaluation set, we sum the frequencies of all types within each paradigm and bucket paradigms based on frequency. We randomly select evaluation paradigms such that all 10 buckets contribute at least 10 paradigms each. For all selected paradigms, any paradigms from the same clan are also selected, allowing us to assume that the paradigms included in the evaluation set are independent of those that are not included. Paradigms with only a single type are discarded, as these are not interesting for analysis. Our resulting EVAL set contains 1,036 words from 91 paradigms and a great deal of ambiguity at all levels of abstraction (see Table 1). Because we prohibit paradigms from entering EVAL without the rest of their clan, EVAL also exhibits the desirable property of reflecting a generally realistic distribution of ambiguity: 36% of its vocabulary are lemma ambiguous as compared to 39% for the entire corpus.

[3]We remove the rarely used diacritics and *Alif/Ya* normalize (El Kholy and Habash, 2012).

[4]We exclude words from the embedding model and evaluation set if they either cannot be analyzed by SAMA, only receive proper noun analyses, or if they do not also occur in the larger Arabic Gigaword corpus (Parker et al., 2011). This controls for many idiosyncrasies.

	Count	Ambiguous	Non-derivationally Ambiguous
Clan	49	18	5
Family	55	24	11
Paradigm	91	60	14
Subparadigm	116	48	6
Word	1,036	372	85

Table 1: Statistics from the EVAL set. Morphological structures by level of abstraction. Ambiguous structures contain at least one lemma ambiguous form. Non-derivationally ambiguous structures contain at least one coincidentally lemma ambiguous form.

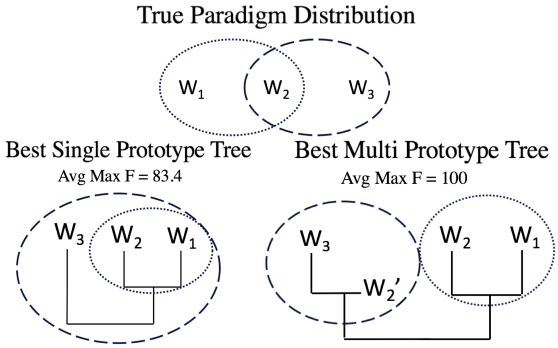

Figure 2: Best clustering strategies for two paradigms–dotted versus dashed ovals–given single or multi prototype vocabulary representations.

3.2 Approach and Evaluation Metric

We build single and multi prototype representations of the entire vocabulary, then examine how well they reflect the paradigms in EVAL. Each representation can be thought of as a tree where each word is a leaf at depth 0, i.e., W_1, W_2, and W_3 in Figure 2. Descending down the tree, words are clustered with other words' branches at subsequent depths until the clustering algorithm finishes or the root is reached where all words in the vocabulary are clustered together. All trees use some model of word similarity to guide clustering. In multi prototype representations, a word's leaf prototype at depth 0 can be copied and grafted onto other words' branches at non-zero depths before those branches are clustered to its own. Such is the case of W_2, which is copied as W_2' at depth 1 of W_3's branch before W_3's branch connects to W_2's. This enables partially overlapping paradigms to be modeled, like those in Figure 2.

We evaluate the trees via average maximum F-score. For each word in EVAL, we descend from its leaf, at each depth calculating an F-score for the overlap between the words that have been clustered to the leaf's branch so far and the leaf word's known paradigm mates, i.e., the set of words sharing at least one lemma with the leaf. Thus, paradigms are soft clusters in our representation, in that, for each word in a paradigm, its set of proposed paradigm mates need not be consistent with any of its proposed paradigm mates' sets of proposed paradigm mates. We then take the best F-score for each leaf word in EVAL, regardless of the depth level at which it was achieved, and average these maximum F-scores. This reflects how cohesively paradigms are represented in the tree.[5] Additionally, we report the average depth at which templatic and concatenatively related paradigm mates are added.

Because we evaluate via average maximum F-score, this metric represents the potential performance of any given model. Future work will address predicting the depth level where average maximum F-score is achieved for a given leaf word via rule-based and/or empirical techniques that have proven successful for related tasks (Narasimhan et al., 2015; Soricut and Och, 2015; Cao and Rei, 2016; Bergmanis and Goldwater, 2017; Sakakini et al., 2017).

3.3 Word Similarity Models

We use the following word similarity models for clustering words in single and multi prototype tree representations.

LEVENSHTEIN The LEVENSHTEIN baseline uses only orthographic edit distances to form a multi prototype tree. At each depth level i, the branch will include every word which has an edit distance of i when compared to the leaf. Transitivity does not hold in this model, as words x and y could be in each other's depth 1 branch, but the fact that z is in y's depth 1 branch does not imply its inclusion in x's depth 1 branch. If the edit distance between x and z is greater than 1, copies, or additional prototypes must be made of x and y. Because morphology involves complicated processes that cannot be explained merely via orthographic similarity, we predict this model will perform poorly. Still, this baseline is useful to ensure that other models are learning something

[5] To control for idiosyncratic paradigms, we calculated a macro F-score averaged over the average maximum F-scores of individual paradigms, though we do not report this as results were not significantly different.

from words' in-context behavior or out-of-context morphological knowledge beyond what can be superficially induced from edit distances.

DELEX We use a de-lexicalized (DELEX) morphological analyzer to predict morphological relatedness. The analyzer covers all MSA closed-class affixes and clitics and their allowed combinations in open class parts-of-speech (POS); however there is no information about stems and lemmas in the model.[6] The affixes and clitics and their compatibility rules were extracted from SAMA (Graff et al., 2009). They are relatively cheap to create for any DA or other languages. The independent, expensive component of SAMA is the information regarding stems and lemmas, which we used to form our evaluation set. We are inspired by Darwish (2002), who demonstrated the creation of an Arabic shallow analyzer in one day. Our approach can be easily extended to DA at least in a similar manner to Salloum and Habash (2014).

To determine if two MSA words are possibly in the same paradigm, we do the following: (1) we use the analyzer to identify all potential stems with corresponding POS for each word (these stems are simply the leftover string after removing any prefixal and suffixal strings which match a prefix-suffix combination deemed compatible by SAMA); (2) each stem is deterministically converted into an orthographic root as per Eskander et al. (2013) by removing Hamzas (the set of letters representing the glottal stop phoneme, i.e., ء *'*, أ *Â*, إ *Ǎ*, آ *Ā*, ؤ *ŵ*, ئ *ŷ*), long vowels (ا *A*, ي *y*, و *w*, ى *ý*), and reducing geminate consonants (e.g., ردد *rdd* → رد *rd*); (3) two words are determined to be possibly from the same paradigm if there exists a possible orthographic root–POS analysis shared by both words.

DELEX builds a multi prototype tree with a maximum depth of 1. For each leaf word, it uses the above algorithm to identify all words in the vocabulary which can possibly share a paradigm with the leaf word, and grafts them into the branch. Hence, a word can belong to more than one hypothesized paradigm. Because DELEX has access to valuable morphological knowledge, we predict it will be a competitive baseline. Furthermore, it should produce nearly perfect recall, only missing rare exceptional forms, e.g., broken plurals that introduce new consonants such as برامج *brAmj*, 'programs', the plural of برنامج *brnAmj*, 'program'. We expect its precision to be weak because it lacks lexical or stem-pattern information, leading to rampant clustering of derivationally related and unrelated forms. For example, a word like جائزة *jAŷzħ*, 'prize' (true root ج و ز *j w z*) receives the orthographic root ج ز *j z* (long vowel, hamza letter, and suffix are dropped), which clusters it with unrelated forms such as جزء *jz'*, 'part' (true root ج ز ء *j z '*), and جز *jz*, 'shearing' (true root ج ز ز *j z z*).

Word Embedding Models (W2V, FT, and FT+) We use different word embedding models to build single prototype representations of the vocabulary via binary hierarchical clustering (Müllner et al., 2013). In order to analyze the effects of data sparsity, we do not impose a minimum word frequency count, but learn vectors for the entire vocabulary. At depth 0, we consider each leaf word to be its own branch. Descending down the tree, we iteratively join the closest two branches based on Ward distance (Ward Jr, 1963). Joined branches are represented by the centroid of their component words' vectors (though, as in other models, we do not include the leaf word as a match when calculating average maximum F-score). We continue iterating until only a single root remains containing the entire vocabulary.

These trees are single prototype because the input embeddings only provide one vector for each word, regardless of whether or not it is ambiguous in any way. While this is a limitation for these models,[7] existing multi prototype word embeddings generally model sense ambiguity, which is easier to capture (though harder to evaluate) given the unsupervised settings in which embeddings are typically trained (Reisinger and Mooney, 2010; Huang et al., 2012; Chen et al., 2014; Bartunov et al., 2016). Adapting multi prototype embed-

[6] The system includes 15 basic prefixes/proclitics (ال *Al*, ا *A*, ب *b*, ف *f*, في *fy*, ك *k*, ل *l*, لا *lA*, ما *mA*, ن *n*, س *s*, ت *t*, و *w*, ي *y*, and ϕ) in 84 unique combinations; and 30 suffixes/enclitics (ا *A*, ان *An*, ات *At*, ه *h*, ها *hA*, هم *hm*, هما *hmA*, هن *hn*, ك *k*, كم *km*, كما *kmA*, كن *kn*, ن *n*, نا *nA*, ني *ny*, ة *ħ*, ت *t*, تا *tA*, تان *tAn*, تم *tm*, تما *tmA*, تن *tn*, تي *ty*, تين *tyn*, و *w*, وا *wA*, ون *wn*, ي *y*, ين *yn* and ϕ) in 193 unique combinations.

[7] A single prototype oracle that always correctly maps non-lemma ambiguous words to their paradigm and maps lemma ambiguous words only to their largest possible paradigm scores 97% (92% specifically on lemma ambiguous types). This represents the best possible performance for single prototype models.

dings to model lemma ambiguities is non-trivial, especially without lots of supervision. We leave this for future work.

Because trees built from word embeddings are all constructed via the same binary clustering algorithm, the depths at which templatic and concatenatively inflected paradigm mates are joined in Table 2 are comparable vertically across W2V, FT, and FT+ as well as horizontally. However, the multi prototype trees are shorter and fatter, such that the templatic and concatenative average join depths are only comparable horizontally with each other, i.e., within the same model.

W2V The Gensim implementation of WORD2VEC (Mikolov et al., 2013a; Řehůřek and Sojka, 2010) uses the SkipGram algorithm with 200 dimensions and a context window of 5 tokens on either side of the target word. As this does not have access to any subword information and is specifically designed for semantics, not morphology, we predict that it will not perform well in our evaluation.

FT We train a FASTTEXT (Bojanowski et al., 2016) implementation with the same parameters as W2V, except a word's vector is the sum of its SkipGram vector and that of all its component character n-grams between length 2 and 6. Since short vowels are not written, many Arabic affixes are only one character. With FASTTEXT bookending words with start/end symbols in its internal representation, outermost single-letter affixes are functionally two characters. By inducing knowledge of such affixes, these character n-gram parameters outperform the language agnostic range of 3 to 6 proposed by Bojanowski et al. (2016).

With the ability to model how subword strings behave in context, FT should outperform both LEVENSHTEIN and W2V, though without access to scholar seeded knowledge of morphological structures, it is difficult to predict how FT will compare with DELEX. Errors may arise from clustering words based on affixes indicative of syntactic behavior instead of the stem, which indicates paradigm membership. Also, if the word is infrequent and contains no semantically distinct subword string with higher frequency, the embeddings will be noisy. Frequency and noise also interact with the *hubbiness*, or crowdedness of the embedding region, as rural regions will require less precision in the vectors to cluster well, whereas there is little room for noise in crowded urban regions where many similar but morphologically unrelated words could interfere.

FT+ We build another FT model by concatenating the vectors learned from two variant FT models, one with the normal window size of 5 and one with a narrow window size of 1. Both are trained on a preprocessed corpus where phrases have been probabilistically identified in potentially unique distributions over multiple copies of each sentence, as described in Erdmann et al. (2018).[8] This technique attempts to better model syntactic cues–which are better encoded with narrow context windows (Pennington et al., 2014; Trask et al., 2015; Goldberg, 2016; Tu et al., 2017)–while avoiding treating non-compositional phrases as compositional, and also learning from multiple, potentially complementary phrase-chunkings of every sentence. By combining these sources of information, FT+ is designed to learn more meaningful vectors without requiring additional data. We predict it will uniformly outperform FT by reducing noise in the handling of sparse forms like infrequent inflections–a hallmark of morphologically rich languages.

FT+&DELEX We make unique copies for each leaf word's branch extending all the way to the root in the single prototype FT+ tree. Then, for each leaf word, at every depth of its branch copy, we use DELEX to prune any words which could not share an orthographic root with the leaf word. Pruning is local to that branch copy, and does not affect the branch copies of paradigm mates which had originally been proposed by FT+ before making branch copies. This makes FT+&DELEX a multi-prototype model. After pruning, the F-score is recalculated for each depth of each leaf word's branch and a new average maximum F-score is reported. Because FT+ encodes information regarding the in-context behavior of words, it is quite complementary to the out-of-context morphological knowledge supplied by DELEX. We predict this model will outperform all others.

[8] For control, we compared every possible combination of narrow and wide window sizes (1 or 5), dimension sizes (200 or 400), and techniques for phrase identification (none, deterministic (Mikolov et al., 2013b), and probabilistic (Erdmann et al., 2018)), but none approached the performance achieved with the parameters used in FT+.

Word Similarity Model	Multi Prototype	Averaged Scores			Join Depth	
		Max F-Score	Precision	Recall	Concat	Temp
LEVENSHTEIN	✓	22.0	35.5	23.4	3.5	4.1
DELEX	✓	52.9	41.6	99.3	1.0	1.0
W2V		2.1	6.7	28.1	17.1	17.4
FT		39.2	66.0	44.2	13.7	16.8
FT+		50.2	71.8	52.9	13.3	16.4
FT+&DELEX	✓	**71.5**	**74.0**	**81.3**	**13.3**	**16.4**

Table 2: Scores for clustering words with their paradigm mates in tree representations built from different models of word similarity. Scores are calculated as described in Section 3.2, with precision and recall extracted from the depth that maximizes F and then averaged over all words in EVAL. Join depths refer to the average depth at which templatic or concatenatively related paradigm mates are added to the branch.

4 Results and Discussion

The results in Table 2 provide strong evidence in support of our hypotheses. The only model performing worse than the LEVENSHTEIN edit distance baseline is W2V, which only understands the in-context, semantic behavior of words. By being able to learn morphological knowledge from in-context behavior of subword strings, FT greatly improves over both W2V and LEVENSHTEIN, demonstrating that it learns far more than can be inferred from out-of-context subword strings, i.e., edit distance, or in-context distributional semantic knowledge without any morphology, i.e., W2V. As predicted, FT+ improves uniformly over FT in all categories, presumably by reducing noise in the vectors of infrequent inflections. Interestingly, with no access to subword information, W2V performs equally poorly on both templatic and concatenatively related paradigm mates, whereas FT and FT+ greatly improve on concatenative mates, but not templatic ones. This is likely because FT and FT+ can identify patterns in subword strings, but not in non-adjacent characters.

DELEX's strong baseline performance demonstrates that simple, out-of-context, de-lexicalized knowledge of morphology is sufficient to outperform the best word embedding model that only learns from words' in-context behaviors. However, given the complementarity between DELEX's knowledge and the information FT+ can learn, it is not surprising that the combination of these techniques, FT+&DELEX, far outperforms either system individually.

Specific Examples We discuss a number of examples that illustrate the variety in the behavior and complementarity of rule-based DELEX, embedding-based FT+, and the combined FT+&DELEX models. For each example, we specify the strength of the maximum F-score for the three models as such:[9] $strength^{\text{DELEX}}$+$strength^{\text{FT+}}$→$strength^{\text{FT+\&DELEX}}$, e.g., LOW+MID→HI denotes poor DELEX and mediocre FT+ performance on a word, yielding high performance in the combined model.

- جائزة *jAŷzħ*, 'prize' (LOW+HI→HI)
  This word has high orthographic root ambiguity since its second morphological root radical is a Hamza. This results in matching words with unrelated true roots like جزء *jz'*, 'part' and جز *jz*, 'shearing' under DELEX. It also has high root fertility, in that different paradigms can come from the same true root, like جَائِز *jAŷz*, 'permissible', further challenging DELEX. FT+ does relatively better, capturing the word's other inflections, even the broken plural جوائز *jwAŷz*, as their in-context behavior is similar to جائزة *jAŷzħ*. Interesting recall errors by FT+ include semantically and orthographically similar فائزة *fAŷzħ*, 'winner[fem.sing]'. The combination yields a perfect F-score.
- يهرعون *yhrςwn*, 'they rush' (HI+LOW→HI)
  This word has an unambiguous orthographic root with no root fertility, resulting in a perfect F-score for DELEX. However, FT+ misses several inflections such as نهرع *nhrς*, 'we rush', and وهرعت *whrςt*, 'and I/you/she rushed'. FT+ also makes many semantically and/or syntactically similar precision errors: يسرعون *ysrςwn*, 'they hurry', يصارعون *ySArςwn*, 'they wrestle', and يقرعون *yqrςwn*, 'they ring (a bell)'. The combination leads to a perfect F-score.
- ديناميكي *dynAmyky*, 'dynamic' (HI+HI→HI)
  This word has an unambiguous orthographic

[9] The strength designation HI is used for F-scores above 75%, LOW for scores below 25%, and MID for the rest.

root based on a foreign borrowing and relatively unique semantics and subword strings. Thus, it achieves a perfect F-score in all three models.

- انتشروا *AntšrwA*, 'they spread out' (MID+MID→HI)
  This word has high orthographic root ambiguity (and, incidentally, fertility) due to the presence of ن *n* and ت *t*, which could belong to a root, template, or prefix. This leads to a 63% F-score under DELEX with many precision errors: انتشاره *AntšArh*, 'his spreading out', ونتشاور *wntšAwr*, 'we discuss', and نتشارك *ntšArk*, 'we collaborate'. FT+ scores only 47%, proposing semantically related but morphologically unrelated or only derivationally related forms: e.g., منتشر *mntšr*, 'spread out' (adjective), and تمركزوا *tmrkzwA*, 'they centralized' (antonym). This semantic knowledge however, complements DELEX's knowledge, such that the combination is almost perfect (98%).

- كفء *kf'*, 'efficient' (LOW+LOW→LOW)
  While 17% of words are LOW in DELEX and 28% in FT+, only 4% are LOW in FT+&DELEX. This word exemplifies that 4%, occupying the gap between DELEX's knowledge and FT+'s. It has an extremely ambiguous orthographic root due to the true root containing a Hamza and the first letter being interpretable as a proclitic or root radical. Thus, DELEX achieves 2% F. FT+ is only slightly better (5%). It is likely that this word's low frequency is the main contributor to its noisy embedding, as it only appears once in our corpus. The combination F-score is thus, only 11%.

5 Related Work

This work builds on several others addressing word embeddings and computational morphology.

Word Embeddings Word embeddings are trained by predicting either a target word given its context (Continuous Bag of Words) or elements of the context given a target (SkipGram) in unannotated corpora (Mikolov et al., 2013a), with the learned vectors modeling how words relate to each other. Embeddings have been adapted to incorporate word order (Trask et al., 2015) or subword information (Bojanowski et al., 2016) to motivate the learned vectors to specifically capture syntactic, morphological, or other similarities.

Word embeddings are generally *single prototype* models, in that they learn one vector for each word, which can be problematic for ambiguous forms (Reisinger and Mooney, 2010; Huang et al., 2012; Chen et al., 2014). Bartunov et al. (2016) propose a *multi prototype* model that learns distinct vectors for distinct meanings of types based on variation in the contexts within which they appear. Gyllensten and Sahlgren (2015), argue that single prototype embeddings actually can model ambiguity because the defining characteristics of a word's different meanings typically manifest in different dimensions of the highly dimensional vector space. They find ambiguous words' relative nearest neighbors in a relative neighborhood graph often correlate with distinct meanings. Such works however, deal with sense ambiguity, or abstract semantic distinctions between different usages of a word with potentially the same morpho-syntactic properties and core meaning. Evaluation usually requires linking to large semantic databases which, for Arabic, are still underdeveloped (Black et al., 2006; Badaro et al., 2014; Elrazzaz et al., 2017).

Computational Morphology This field of study includes rule-based, machine learning, and hybrid approaches to modeling morphology. The traditional approach is to hand write rules to identify the morphological properties of words (Beesley, 1998; Khoja and Garside, 1999; Habash and Rambow, 2006; Smrž, 2007; Graff et al., 2009; Habash, 2010). These can be used for out-of-context analysis–which SAMA (Graff et al., 2009) performs for MSA–or they can be combined with machine learning approaches that leverage information from the context in which a word appears. MADAMIRA (Pasha et al., 2014), for example, is trained on an annotated corpus to disambiguate SAMA's analyses based on the surrounding sentence.

Other systems use machine learning without rules. They can train on annotated data, like Faruqui et al. (2016) who learn morpho-syntactic lexica from a small seed, or they can learn without supervision, like Luo et al. (2017) who induce "morphological forests" of derivationally related words by predicting suffixes and prefixes based on the vocabulary alone. Some approaches seek to be language independent. MORFESSOR (Creutz and Lagus, 2005), for instance, segments words based on unannotated text. However, it deter-

ministically produces context-irrelevant segmentations, causing error propagation in languages like Arabic, characterized by high lexical ambiguity (Saleh and Habash, 2009; Pasha et al., 2014). A few systems have incorporated word embeddings to perform segmentation (Narasimhan et al., 2015; Soricut and Och, 2015; Cao and Rei, 2016), with some attempting to model and analyze relations between underlying morphemes as well (Bergmanis and Goldwater, 2017; Sakakini et al., 2017), though none of these distinguish between inflectional and derivational morphology. Eskander et al. (2016b) propose another segmentation system using Adaptor Grammars for six typologically distinct languages. Snyder and Barzilay (2010) actually use multiple languages simultaneously, finding the parallels between them useful for disambiguation in morphological and syntactic tasks.

Our work is closely related to Avraham and Goldberg (2017), who train embeddings on a Hebrew corpus with disambiguated morpho-syntactic information appended to each token. Similarly, Cotterell and Schütze (2015) "guide" German word embeddings with morphological annotation, and Gieske (2017) use morphological information encoded in word embeddings to inflect German verbs. For Arabic, Rasooli et al. (2014) induce paradigmatic knowledge from raw text to produce unseen inflections, and Eskander et al. (2013) identify *orthographic roots* and use them to extract features for paradigm completion given annotated data. While we adopt the concept of approximating the linguistic root with an orthographic root, we do not use annotated data where the stem has already been determined as in Eskander et al. (2013). Thus, we generate all possible orthographic roots for a given word instead of just one, as discussed in Section 3.3.

Sakakini et al. (2017) provide an alternative unsupervised technique for extracting roots in Semitic languages, however, we chose to adopt the orthographic root concept instead for several reasons. Firstly, despite performing comparably with other empirical techniques, Sakakini et al. (2017)'s root extractor is not extremely accurate. While our implementation generates potentially multiple orthographic roots with imperfect precision, the near perfect recall is useful for pruning without propogating error. A major reason why we find DELEX and FT+ to complement one another is the independence of the orthographic root extraction rules and the distributional statistics leveraged by word embeddings. Sakakini et al. (2017)'s root extractor however, depends on embeddings to identify roots. Furthermore, their root extractor cannot be used to generate multi prototype models as it only produces one root per word. Finally, despite orthographic roots' dependance on hand written rules, we show that these rules are very few, such that adapting Sakakini et al. (2017)'s root extractor to a new language or dialect would not necessarily require any less effort than writing new rules.

6 Conclusion and Future Work

In this work, we demonstrated that out-of-context, rule-based knowledge of morphological structure, even in minimal supply, greatly complements what word embeddings can learn about morphology from words' in-context behaviors. We discussed how Arabic's morphological richness and many forms of ambiguity interact with different word similarity models' ability to represent morphological structure in a paradigm clustering task. Our work quantifies the value of leveraging subword information when learning embeddings and the further value of noise reduction techniques targeting the sparsity caused by complex morphology. Our best performing model uses out-of-context rules to prune unlikely paradigm mates suggested by our best embedding model, achieving an F-score of 71.5% averaged over our evaluation vocabulary. Our results inform how one would most cost effectively construct morphological resources for DA or similarly under resourced, morphologically complex languages.

Our future work will target templatic morphological processes which still challenge our best model, requiring knowledge of patterns realized over non-adjacent characters. We will also address errors due to ambiguity, either by adapting multi prototype embedding models to capture morphological ambiguity, including knowledge of paradigm structure in our de-lexicalized rules, or by using disambiguated lemma frequencies to model ambiguity probabilistically. In applying this work to DA, we will additionally need to address the issue of noisy, unstandardized spelling. We will also investigate different knowledge transfer techniques to leverage the many resources available for MSA.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16.

Faisal Al-Shargi, Aidan Kaplan, Ramy Eskander, Nizar Habash, and Owen Rambow. 2016. Morphologically annotated corpora and morphological analyzers for Moroccan and Sanaani Yemeni Arabic. In *10th Language Resources and Evaluation Conference (LREC 2016)*.

Sarah Alkuhlani and Nizar Habash. 2011. A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, Oregon, USA.

Amjad Almahairi, Kyunghyun Cho, Nizar Habash, and Aaron C. Courville. 2016. First result on Arabic neural machine translation. *CoRR*, abs/1606.02680.

Oded Avraham and Yoav Goldberg. 2017. The interplay of semantics and morphology in word embeddings. *arXiv preprint arXiv:1704.01938*.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. A large scale Arabic sentiment lexicon for Arabic opinion mining. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138.

Kenneth Beesley. 1998. Arabic morphology using only finite-state operations. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 50–7, Montereal.

Toms Bergmanis and Sharon Goldwater. 2017. From segmentation to analyses: A probabilistic model for unsupervised morphology induction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 337–346.

William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. 2006. Introducing the Arabic wordnet project. In *Proceedings of the third international WordNet conference*, pages 295–300. Citeseer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Kris Cao and Marek Rei. 2016. A joint model for word embedding and word morphology. *arXiv preprint arXiv:1606.02601*.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic Dialects. In *Proceedings of EACL*, Trento, Italy. EACL.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. *CoRR*, abs/1706.09031.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292.

Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology.

Kareem Darwish. 2002. Building a shallow Arabic morphological analyzer in one day. In *Computational Approaches to Semitic Languages, an ACL'02 Workshop*, pages 47–54, Philadelphia, PA.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for English–Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

Mohammed Elrazzaz, Shady Elbassuoni, Khaled Shaban, and Chadi Helwe. 2017. Methodical evaluation of Arabic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–458, Vancouver, Canada.

Alexander Erdmann, Nizar Habash, Dima Taji, and Houda Bouamor. 2017. Low resourced machine translation via morpho-syntactic modeling: The case of dialectal arabic. In *Proceedings of MT Summit 2017*, Nagoya, Japan.

Alexander Erdmann, Nasser Zalmout, and Nizar Habash. 2018. Addressing noise in multidialectal word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 558–565.

Ramy Eskander, Nizar Habash, and Owen Rambow. 2013. Automatic extraction of morphological lexicons from morphologically annotated corpora. In

Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1032–1043.

Ramy Eskander, Nizar Habash, Owen Rambow, and Arfath Pasha. 2016a. Creating resources for Dialectal Arabic from a single annotation: A case study on Egyptian and Levantine. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3455–3465, Osaka, Japan.

Ramy Eskander, Owen Rambow, and Tianchun Yang. 2016b. Extending the use of adaptor grammars for unsupervised morphological segmentation of unseen languages. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 900–910.

Manaal Faruqui, Ryan McDonald, and Radu Soricut. 2016. Morpho-syntactic lexicon generation using graph-based semi-supervised learning. *Transactions of the Association for Computational Linguistics*, 4:1–16.

Sharon Gieske. 2017. Inflecting verbs with word embeddings: A systematic investigation of morphological information captured by German verb embeddings. Master's thesis, University of Amsterdam.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *J. Artif. Intell. Res.(JAIR)*, 57:345–420.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic morphological analyzer (SAMA) version 3.1. *Linguistic Data Consortium LDC2009E73*.

Amaru Cuba Gyllensten and Magnus Sahlgren. 2015. Navigating the semantic horizon using relative neighborhood graphs. *arXiv preprint arXiv:1501.02670*.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A morphological analyzer for Egyptian Arabic. In *Proceedings of the twelfth meeting of the special interest group on computational morphology and phonology*, pages 1–9. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2006. MAGEAD: a morphological analyzer and generator for the Arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 681–688. Association for Computational Linguistics.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Go Inoue, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Joint prediction of morphosyntactic categories for fine-grained Arabic part-of-speech tagging exploiting tag dictionary information. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 421–431, Vancouver, Canada. Association for Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a corpus for Palestinian Arabic: a preliminary study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27.

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A Morphologically Annotated Corpus of Emirati Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Salam Khalifa, Sara Hassan, and Nizar Habash. 2017. A morphological analyzer for Gulf Arabic verbs. *WANLP 2017 (co-located with EACL 2017)*, page 35.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2016. Yamama: Yet another multi-dialect Arabic morphological analyzer. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 223–227.

Shereen Khoja and Roger Garside. 1999. Stemming Arabic text. *Lancaster, UK, Computing Department, Lancaster University*.

Jiaming Luo, Karthik Narasimhan, and Regina Barzilay. 2017. Unsupervised learning of morphological forests. *arXiv preprint arXiv:1702.07015*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Daniel Müllner et al. 2013. fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. *Journal of Statistical Software*, 53(9):1–18.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *arXiv preprint arXiv:1503.02335*.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2011. Arabic Gigaword Fifth Edition. LDC catalog number No. LDC2011T11, ISBN 1-58563-595-2.

Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *In Proceedings of LREC*, Reykjavik, Iceland.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Mohammad Sadegh Rasooli, Thomas Lippincott, Nizar Habash, and Owen Rambow. 2014. Unsupervised morphology-based vocabulary expansion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1349–1359.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Tarek Sakakini, Suma Bhat, and Pramod Viswanath. 2017. Fixing the infix: Unsupervised discovery of root-and-pattern morphology. *arXiv preprint arXiv:1702.02211*.

Ibrahim M. Saleh and Nizar Habash. 2009. Automatic extraction of lemma-based bilingual dictionaries for morphologically rich languages. In *Proceedings of MT Summit*, Ottawa, Canada.

Wael Salloum and Nizar Habash. 2014. ADAM: Analyzer for Dialectal Arabic Morphology. *Journal of King Saud University-Computer and Information Sciences*, 26(4):372–378.

Otakar Smrž. 2007. *Functional Arabic Morphology. Formal System and Implementation*. Ph.D. thesis, Charles University, Prague.

Benjamin Snyder and Regina Barzilay. 2010. Climbing the tower of Babel: Unsupervised multilingual learning. In *Proceedings of the International Conference on Machine Learning (ICML-10)*, Haifa, Israel.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637.

Andrew Trask, David Gilmore, and Matthew Russell. 2015. Modeling order in neural word embeddings at scale. *arXiv preprint arXiv:1506.02338*.

Lifu Tu, Kevin Gimpel, and Karen Livescu. 2017. Learning to embed words in context for syntactic tasks. *arXiv preprint arXiv:1706.02807*.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Nasser Zalmout, Alexander Erdmann, and Nizar Habash. 2018. Noise-robust morphological disambiguation for dialectal Arabic. In *Proceedings of the 16th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL18)*, New Orleans, Louisiana, USA.

Nasser Zalmout and Nizar Habash. 2017. Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 715–724.

Inès Zribi, Mariem Ellouze, Lamia Hadrich Belguith, and Philippe Blache. 2017. Morphological disambiguation of Tunisian dialect. *Journal of King Saud University-Computer and Information Sciences*, 29(2):147–155.