Morphological Word Embeddings for Arabic Neural Machine Translation in Low-Resource Settings

Pamela Shapiro
Johns Hopkins University
pshapiro@jhu.edu

Kevin Duh
Johns Hopkins University
kevinduh@cs.jhu.edu

Abstract

Neural machine translation has achieved impressive results in the last few years, but its success has been limited to settings with large amounts of parallel data. One way to improve NMT for lower-resource settings is to initialize a word-based NMT model with pretrained word embeddings. However, rare words still suffer from lower quality word embeddings when trained with standard word-level objectives. We introduce word embeddings that utilize morphological resources, and compare to purely unsupervised alternatives. We work with Arabic, a morphologically rich language with available linguistic resources, and perform Ar-to-En MT experiments on a small corpus of TED subtitles. We find that word embeddings utilizing subword information consistently outperform standard word embeddings on a word similarity task and as initialization of the source word embeddings in a low-resource NMT system.

1 Introduction

Neural machine translation (Bahdanau et al., 2014; Sutskever et al., 2014) has recently become the dominant approach to machine translation. However, the standard encoder-decoder models with attention have been shown to perform poorly in low-resource settings (Koehn and Knowles, 2017), a problem which can be alleviated by initialization of parameters from an NMT system trained on higher-resource languages (Zoph et al., 2016). An alternative way to initialize parameters in a low-resource NMT setup is to use pretrained monolingual word embeddings, which are quick to train and readily available for many languages.

There is a large body of work on word embeddings. Popular approaches include `word2vec` (Mikolov et al., 2013a) and `GloVe` (Pennington et al., 2014). These have been shown to perform well in word similarity tasks and a variety of downstream tasks. However, they have been primarily evaluated on English. The learned representations for rare words are of low quality due to sparsity. For morphologically rich languages, we may want word embeddings that also consider morphological information, to reduce sparsity in word embedding training.

Previous work on morphological word embeddings has shown improvements on word similarity tasks, but has not been evaluated on downstream NMT tasks. Our contribution is two-fold:

1. We adapt `word2vec` to utilize lemmas from a morphological analyzer,[1] and show improvements on a word similarity task over a state-of-the-art unsupervised approach to incorporating morphological information based on character n-grams (Bojanowski et al., 2017).

2. We experiment with Arabic-to-English NMT on the TED Talks corpus. Our results demonstrate that incorporating some form of morphological word embeddings into NMT improves BLEU scores and outperforms the conventional approaches of using standard word embeddings, random initialization, or byte-pair encoding (BPE).

2 Neural Machine Translation

We follow recent work in neural machine translation, using a standard bi-directional LSTM encoder-decoder model with the attention mechanism from Luong et al. (2015). We describe below other work in NMT that has tried to address some of the same issues dealing with settings with limited parallel data, improving translation of morphological complexity, and Arabic NMT.

[1] https://github.com/pamelashapiro/word2vec_morph

Proceedings of the Second Workshop on Subword/Character LEvel Models, pages 1–11
New Orleans, Louisiana, June 6, 2018.

2.1 Low-Resource Settings

Some success has been achieved applying neural machine translation to low-resource settings. Zoph et al. (2016) use transfer learning to improve NMT from low-resource languages into English. They initialize parameters in the low-resource setting with parameters from an NMT model trained on a high-resource language. Nguyen and Chiang (2017) extend this by exploiting vocabulary overlap in related languages. Similarly, Firat et al. (2016) share parameters between high and low resource languages via multi-way, multilingual NMT.

Other work aims to exploit monolingual data via back-translation (Sennrich et al., 2016a). Imankulova et al. (2017) aim to improve this technique for low-resource settings by filtering generated back-translations with quality estimation. Meanwhile, He et al. (2016) use a reinforcement learning approach to learn from monolingual data.

Our approach is similar to those utilizing transfer learning, but we initialize on the source side with monolingual word embeddings, which is relatively simple to implement and low-cost to train. Di Gangi and Marcello (2017) experiment with monolingual word embeddings as we do, but they merge external monolingual word embeddings with the embeddings learned by an NMT system. We simply use word embeddings as initialization, and we instead focus on exploring how morphological word embeddings can help in this setup.

2.2 Incorporating Morphology

Some research has aimed to incorporate morphological information into NMT systems. Byte-Pair Encoding (BPE) segments words into pieces by merging character sequences based on frequency (Sennrich et al., 2016b), and these sequences of word pieces are translated. BPE become standard practice. However, it is unclear how much data is necessary for it to be beneficial. In our experiments, BPE performs worse than initializing with any of the word embeddings for our dataset.

Character-level NMT has recently become popular as well (Ling et al., 2015b; Costa-jussà and Fonollosa, 2016; Lee et al., 2017). Their work aims to implicitly learn morphology by building neural network architectures over characters. We also compare to a character-level NMT system in our experiments.

Additionally, Dalvi et al. (2017) add morphological information into the decoder, following work from Belinkov et al. (2017) that showed that the encoder already learns more morphological information than the decoder. Our work differs in that we are focusing on incorporating morphological information into the source side. Moreover, Belinkov et al. (2017) works with higher-resource datasets. It is possible that in lower-resource settings, it will still be helpful to incorporate morphological information into the encoder.

2.3 Arabic NMT

Almahairi et al. (2016) produce the first results of neural machine translation on Arabic. They find that preprocessing of Arabic as used in statistical machine translation is helpful. They normalize the text, removing diacritics and normalizing inconsistently typed characters, and they tokenize according the Penn Arabic Treebank (ATB) scheme (Maamouri et al., 2004), separating all clitics except for definite articles. We normalize as such, but do not use ATB tokenization, instead using the default tokenization in Moses (Koehn et al., 2007). We do this to focus on embeddings for words and to facilitate generalization to other languages. Additionally, Sajjad et al. (2017) explore alternatives to language-specific segmentation in Arabic, finding that BPE performs the best in their scenario.

Note that unlike the previously described work, we are using a dataset of only 2.9 million tokens for training. This is to assess the use of morphological word embeddings in settings with limited parallel data.

3 Morphological Word Embeddings

Morphological word embeddings help improve the quality of pretrained word embeddings for less frequent morphological variants, which is important for morphologically rich and low-resource languages. We outline related work in this section and describe an additional approach of our own.

Some related work has used morphological resources to guide word embeddings. Cotterell and Schütze (2015) use a multi-task objective to encourage word embeddings to reflect morphological tags, working within the log-bilinear model of Mnih and Hinton (2007). Cotterell et al. (2016) use a latent-variable model to adapt existing word embeddings to morphemes. Our additional approach is similar to this vein of work in that it uses morphological resources, but it works within the popu-

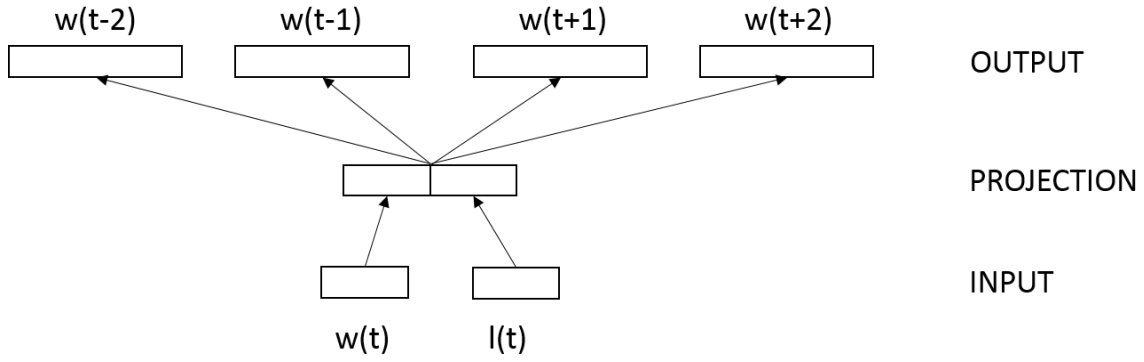

Figure 1: Modified skipgram objective for training `morph` embeddings. Here, w(t) is the current word, l(t) is its lemma, and w(t-2), w(t-1),w(t+1),w(t+2) are neighboring words.

lar `word2vec` skipgram objective (Mikolov et al., 2013a), adding a simple modification to consider a lemma in addition to a word form.

Other work uses purely unsupervised techniques. Luong et al. (2013) segment words using Morfessor (Creutz and Lagus, 2007), and use recursive neural networks to build word embeddings from morph embeddings. Instead of explicit segmentation, `fastText` (Bojanowski et al., 2017) incorporates subword information into the skipgram model by treating a word as a bag of character n-grams. They represent each n-gram of sizes 3-6 with a vector, and each word as a sum of its n-gram vectors. While `fastText` is not explicitly learning morphology, it can be viewed as potentially incorporating morpheme-like subwords.

For simplicity and efficiency, we consider only embeddings in the skipgram family—`fastText`, `word2vec` skipgram, and our modification of the `word2vec` skipgram objective, described in 3.1. There is a large literature on exploiting characters, morphology, and composition for embedding models (Chen et al., 2015; Ling et al., 2015a; Qiu et al., 2014; Wieting et al., 2016; Lazaridou et al., 2013), and a comparison with these different models may be interesting future work.

The usefulness of word embeddings in downstream applications is a question that often needs to be revisited. Many types of morphological or character-level embedding models have been evaluated under various extrinsic metrics, in applications such as language modeling (Kim et al., 2016; Botha and Blunsom, 2014; Sperr et al., 2013), parsing (Ballesteros et al., 2015), part-of-speech tagging (dos Santos and Zadrozny, 2014), and named-entity recognition (dos Santos and Guimarães, 2015; Cotterell and Duh, 2017). Besides the Arabic word similarity dataset, here we also focus on evaluating embeddings at the source side of a machine translation task.

3.1 Modified Skipgram Objective

We assume the availability of a morphological analyzer or lemmatizer that will output a lemma for each word token in a text. We modify the skipgram objective (Mikolov et al., 2013b) to use both word and lemma to predict context words, as illustrated in Figure 1. We learn word vectors and lemma vectors, using their concatenation in the dot product with a context vector in the skipgram objective. So the modified objective we are approximating with negative sampling is now

$$p(w_O|w_I, l_I) = \frac{\exp(v'^T_{w_O}\text{concat}(v_{w_I}, v_{l_I}))}{\sum_{w=1}^{W}\exp(v'^T_{w}\text{concat}(v_{w_I}, v_{l_I}))}$$

Without the lemma part, this objective corresponds to `word2vec`.

Because there may be multiple lemmas associated with a word type, we use a weighted average over lemma vectors in the final vector:

$$w^*_I = \text{concat}(\mathbf{v_{w_I}}, \frac{1}{c(w_I)}\sum_{l_I} c(w_I : l_I) * \mathbf{v_{l_I}})$$

where $c(\cdot)$ is the count of a word or word-lemma pair. When the morphological analyzer cannot produce a lemma, we use the word form itself. We output the vectors associated with individual lemmas as well, which can be used to handle OOV words.

The lemma simplifies a word, removing clitics and some inflectional morphology. While it reduces sparsity of infrequent stems, it also removes potentially useful information. The hope is that by using both word and lemma, we can maintain enough of the benefits of morphology in frequent words while also reducing sparsity in infrequent words. We do some preliminary experiments using just the lemma to predict context words as well, but in preliminary experiments this performed worse, possibly because we lose too much information from morphology.

In future work, we could also try modifying what is predicted as well (i.e. instead of predicting context words, predict lemma or both word and lemma).[2]

4 Arabic Morphology and Resources

We describe here the morphological analyzer we use, as well as prominent features of Arabic morphology that we consider in our analysis.

4.1 Morphological Analyzer

We use a morphological analyzer for Arabic called MADAMIRA (Pasha et al., 2014). MADAMIRA performs rule-based morphological analysis on the form of the word and then uses supervised learning techniques to disambiguate in context. It provides several types of morphological analysis for Arabic. In this work we only use the lemma, though future work could consider utilizing the other morphological information provided.

4.2 Arabic Morphology

One prominent feature of Arabic morphology is that it is rich with clitics, morphemes that syntactically function as words but phonologically function as affixes. Arabic proclitics (prefixes) include articles, conjunctions, and prepositions. Arabic enclitics (suffixes) include object or possessive pronouns. There are also inflectional affixes for number (singular, plural, and dual) and gender (masculine, feminine), and grammatical case endings - though only certain indefinite accusative case endings are visible without diacritics.

Semitic languages such as Arabic also have a substantial amount of non-concatenative morphology. Most stems are formed from a 3-consonant root inserted into a vowelled template, called "templatic morphology." When we are only considering inflectional morphology, as we are in the case of lemmas, we see this most in "broken plurals," which are especially productive in Arabic (as compared to other Semitic languages). A broken plural changes the internal vowelled pattern from the singular, rather than attaching a suffix.

An example of this is the word for "key," mftAH مفتاح, and its plural, mfAtyH مفاتيح, where the root is f-t-H, and the pattern for singular is mCCAC, and for plural is mCACyC.[3] In this case, MADAMIRA would produce the lemma: 1_مِفْتاح for both forms. We hypothesize that the embeddings informed by MADAMIRA will have an advantage on these words, where the morphemes involved cannot be captured by character n-grams.

5 Experiments

We compare three types of embeddings:

- `word2vec`: standard skip-gram word embeddings that only use word information.
- `fastText`: skip-gram embeddings that are sums of vectors representing character n-grams, implicitly incorporating some form of morphological information.
- `morph`: the modified skip-gram word embeddings described in Section 3.1, which rely on a morphological analyzer and lemma embeddings.

The word embeddings inserted into the NMT system are always of dimension 300, and in word similarity experiments, we experiment with dimensions of different sizes. All word embeddings are trained with negative sampling (5 samples), with a window size of 5, a 10^{-4} rejection threshold for subsampling, and 5 iterations. Additional `fastText` parameters are left at the default. We use OpenNMT-py (Klein et al., 2017) for all NMT experiments, with a max sentence size of 80. We use word-level prediction accuracy for model selection. For the BPE baseline, the number of BPE merge operations is 30,000. The hidden layer size is 1024, trained with batch size 80, with Adadelta (Zeiler, 2012) and a dropout rate of 0.2 for 20 epochs with a learning rate of 1.0.

When initializing the encoder with word embeddings, we experiment both with locking the word

[2] Our adaptation of `word2vec` can be used for context-dependent word tags in general, not just lemmas.

[3] We use Buckwalter transliteration (Buckwalter, 2002).

	Normalize Diacritics		Full Normalization	
	Null OOVs	**Handle OOVs**	**Null OOVs**	**Handle OOVs**
word2vec, 150	0.52	NA	0.52	NA
word2vec, 300	0.51	NA	0.53	NA
fastText, 150	0.53	0.55	0.55	0.55
fastText, 300	0.53	0.55	0.54	0.55
morph, 150-150, *word*	0.15	NA	0.15	NA
morph, 150-150, *word+lemma*	0.54	0.55	0.54	0.55
morph, 150-150, *lemma*	**0.59**	**0.60**	**0.59**	**0.60**

Table 1: Spearman coefficient for Arabic word similarity dataset built off of WS353. We list the dimension of the word embedding, and in the case of morph, we list the dimensions of the word part and the lemma part. In the morph system, *lemma* refers to using just the lemma part of the vector to compare similarity, *word* refers to using just the word part, and *word+lemma* refers to using the whole vector.

embeddings throughout training ("*fixed*") and allowing backpropagation through the word embeddings ("*unfixed*"). At test time, words not seen in the MT training data are also initialized with word embeddings, if they were seen in the word embedding training data. Words unseen by either corpus are mapped to the embedding of an <unk> token.

The bitext we use for NMT is a collection of TED subtitles obtained from WIT[3] (Cettolo et al., 2012).[4] This is a collection of monologue speeches from TED talks, covering a wide range of topics such technology, design, and social science. We downloaded the latest XML files (version 2016-04-08) for Arabic and performed subtitle extraction and sentence merging using the WIT[3] scripts. The data is then randomly split at the granularity of talks, with 1939 talks for training, 30 talks for development, and 30 talks for testing.[5] The corresponding sentence/token statistics are shown in Table 2. In this data, 9% of word types and 3% of tokens in the test data were not seen in train.

The monolingual corpus we use for word embeddings is cleaned and tokenized Arabic Wikipedia, consisting of about 80 million tokens, with a vocabulary of around 350k words. The word embeddings are trained on both the monolingual corpus and the source side of the TED training data. The number of lemma types in the monolingual corpus is 672k, and in TED training data is 42k.

5.1 Word Similarity Results

Before running NMT, we first experiment on a word similarity dataset to test the effectiveness of morphology in word embeddings. We compare word2vec, fastText, and variants of our morphological skip-gram in Section 3.1. We experiment with normalizing only diacritics as well as additionally normalizing inconsistently typed characters as in Almahairi et al. (2016), referred to here as "full normalization." We normalize the word similarity dataset accordingly. To ensure that dimensionality is not a major factor, we experiment with various dimensions. We also experiment with just using lemmas to predict, which performs slightly worse than using both word and lemma and taking the lemma part of the vector, though still better than word2vec and fastText.

We evaluate on an Arabic dataset developed by Hassan and Mihalcea (2009) based on the classic *WordSim353* (Finkelstein et al., 2001), as is evaluated on by Bojanowski et al. (2017). We re-run on word2vec and fastText and obtain similar, though not identical, results to Bojanowski et al. (2017). We suspect the differences are due to differences in cleaning and tokenizing Arabic Wikipedia. As is standard for these evaluations, we report Spearman rank coefficient in Table 1.

There are 3 OOV words when normalizing diacritics, and 1 with full normalization, out of 353

Corpus	Sentences	Tokens	Types
Wikipedia	1,751k	79,793k	1,263k
TED, train	175k	2,855k	152k
TED, dev	2k	30k	8k
TED, test	2k	29k	8k

Table 2: Size of corpora, the number of tokens for MT data refers to the source side.

[4] https://wit3.fbk.eu

[5] The data splits are available at http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/.

Model	Average	Δ	3 Runs
random initialization (word)	26.55	-	(26.40, 26.55, 26.70)
random initialization (BPE)	27.80	1.25	(27.64, 27.87, 27.90)
`word2vec`, *fixed*	26.97	0.42	(26.79, 27.00, 27.11)
`word2vec`, *unfixed*	28.38	1.83	(28.25, 28.38, 28.51)
`morph`, *fixed*	28.15	1.60	(27.91, 28.25, 28.29)
`morph`, *unfixed*	28.76	2.21	(28.50, 28.81, 28.96)
`fastText`, *fixed*	28.66	2.11	(28.62, 28.64, 28.71)
`fastText`, *unfixed*	**29.10**	2.55	(28.91, 29.15, 29.24)

Table 3: Corpus-level BLEU on the test set, averaged over 3 runs, with individual runs. Δ is the difference in BLEU between the model vs. random initialization with words as units.

Model	Average	Δ	3 Runs
random initialization (word)	22.85	-	(22.50, 22.90, 23.14)
`word2vec`, *unfixed*	24.89	2.04	(24.76, 24.96, 24.96)
`morph`, *unfixed*	25.49	2.64	(25.20, 25.42, 25.85)
`fastText`, *unfixed*	**25.77**	2.92	(25.49, 25.79, 26.02)

Table 4: BLEU on test sentences that have rare morphological variants. Δ is the difference in BLEU between the model vs. random initialization with words as units.

word pairs. We report results both using zero vectors for OOV and with an attempt to handle OOVs when possible, as done by Bojanowski et al. (2017). To handle OOVs, we run MADAMIRA on the unknown form alone (without the benefit of a context sentence) to get a lemma, and use the lemma vector learned for the corresponding lemma, if it was seen in training, with zeros for the word part.[6]

We see that across normalization schemes and dimensions, `fastText` performs 1-3 points better than `word2vec` in the null OOV setting and 2-4 points better handling OOVs. Using both word and lemma to predict context words performs about the same as `fastText`. However, when we take just the part of the vector corresponded to a weighted average of lemma vectors, it performs 4-6 points better than `fastText`. 2-4 points of this gain can be achieved by just using the lemma to predict context words.

Interestingly, the word part of the `morph` vector performs poorly on word similarity, but still provides some benefit in training. We found that using just the lemma to predict in training performed slightly worse than the lemma part of the vector when using both. It is possible that complementary features are learned in the word part and lemma part of the vector, and that the lemma part corresponds much more closely to semantic similarity.

5.2 Neural Machine Translation Results

We run 3 replicates of experiments with random initializations (re-training word embeddings on each run as well). Results for corpus-level BLEU, calculated using the multi_bleu.sh script from Moses are in provided in Table 3.

BPE outperforms using full words by 1.3 BLEU points (27.80 vs. 26.55). Initializing with `word2vec` results in a 1.8 BLEU point gain over randomly initialized word embeddings. `morph` results in a 0.4 BLEU point gain over `word2vec`, and `fastText` a 0.7 BLEU point gain. Fixing the embeddings consistently performs worse than allowing backpropagation. However, this gap narrows as the BLEU scores of both improve. We also compare to running a NMT system with a CNN over character embeddings in the encoder from Costa-jussà and Fonollosa (2016), which results in a BLEU score of 26.46. [7]

We also perform statistical significance testing via bootstrap resampling, using the `multeval` tool (Clark et al., 2011). The best BLEU are

[6]Note that when attempting to handle OOVs, in the case where we are only normalizing diacritics, we can only recover a lemma vector for 1 of the 3 OOVs while `fastText` is using n-grams to recover something for all 3. In the case of full normalization, both are able to recover a vector.

[7]We use the code from `https://github.com/harvardnlp/seq2seq-attn`, modifying hyperparameters to match our word-level models as closely as possible and using character-level default settings.

28.76 for morph and 29.10 for fastText. Both morph and fastText improve upon word2vec (28.38) with p-values < 0.01. The differences between fastText and morph are not statistically significant.

To see whether trends in BLEU are stronger for sentences containing rarer words with more frequent lemmas, we try filtering test sentences by the ratio of word count to lemma count in the source side of the MT training data. We take sentences with at least one word that has a lemma that is more than 50 times as frequent as the word in training data. Comparing just the unfixed, normalized, word-based versions, we show results for BLEU on filtered sentences in Table 4.

With this heuristic for rare morphological variants, there are 1,376 rare morphological variants out of the 7,345 words that are in the intersection of train and test source data. The heuristic pulls out 1,038 out of 1,982 test sentences to evaluate on. morph results in a 0.6 BLEU point gain over word2vec, and fastText a 0.88 BLEU point gain.

Because of corpus-level BLEU's limitations in characterizing translation quality with respect to morphological variants at the word level, we also perform a manual analysis of the sentences from each system to inspect improvements that may be due to the various word embeddings. We use multeval (Clark et al., 2011) to inspect the sentences that had the biggest sentence-level BLEU improvement over standard word2vec in the morph and fastText cases at the sentence level and see if there are notable trends. We display the median system's translation in this analysis, as recommended by Clark et al. (2011), though sentences selected here exhibited the phenomena described consistently across multiple runs. Example sentences are shown in Table 5.

In several cases, both morph and fastText systems consistently successfully translate rare or unseen words with morphological variants that are seen more commonly in the word embedding training data, while the word2vec system does not. For instance, in example 1, the word **للتدخلات** (*lltdxlAt*, "of interventions") is never seen in the MT training data. It is only seen rarely in the word embedding training data, 24 times. However, the word stripped of the definite article and the clitic corresponding to "of," i.e. the character n-gram **تدخلات** *tdxlAt*, is seen 657 times in word embedding training data. The lemma, which is shared between singular and plural as well, occurs 6,887 times.

In some cases, the morph system is consistently the only system that successfully translates a rare morphological variant. For instance, in example 2, the morph system translates the word **ابعادا** (*AbEAdA*, "dimensions") correctly, while the other systems do not. It occurs here in the accusative case, which does not appear explicitly in many settings in Arabic. This word form occurs 7 times in the MT training data and 101 times in the monolingual corpus. Meanwhile, the lemma **بُعد_1** occurs 214,297 times in the word embedding data. This is much more frequent than we'd expect to see variants of the word "dimension," because the lemma is also associated with the very frequent word for "after." However, it seems to learn a good representation despite this. It is unclear exactly why fastText does not learn a good representation in any of the three runs although it is possible that with character n-grams, there is conflict with other unrelated words. Note that because the plural is non-concatenative, none of the character n-grams in this word corresponds to the singular.

In other cases, the morphological analyzer cannot provide an analysis for a word, and a rare morphological variant is only translated correctly by fastText. In example 3, while sentence-level BLEU is best in the word2vec version in this case, we see a word that is translated best with fastText, and fails to be translated in the other two systems. The word **ابتلاع** (*AbtlAE*, "swallowing") is only seen as a word itself twice in MT training data and 171 times in the monolingual corpus. However, the 6-gram corresponding to the word is seen 444 times in the word embedding training data as a part of other words. Meanwhile, the morphological analyzer does not provide an analysis. While fastText translates as "swallow" rather than "swallowing," it is better than morph for this word, which consistently fails to translate the word at all.

6 Discussion

Overall, morphologically aware word embeddings (morph and fastText) can help reduce sparsity and improve results on both a word similarity task and a low-resource NMT system when used as initialization. The improvements over standard word embeddings is consistent, and implies that

src	1) .و هكذا فتلك امثلة للتدخلات الايجابية
src-Buckwalter	w hk*A ftlk Amvlp lltdxlAt AlAyjAbyp.
ref	So those are examples of positive interventions.
word2vec	And so these are examples of positive feedback.
morph	And so these are examples for positive interventions.
fastText	And so these are examples of positive interventions.
src	2) انا اخبركم ان هناك ابعادا كثيرة للتطور
src-Buckwalter	AnA Axbrkm An hnAk AbEAdA kvyrp lltTwr.
ref	I'm telling you that there are many dimensions of development.
word2vec	I'm telling you that there's a lot of implications of evolution.
morph	I'm telling you that there are many dimensions for evolution.
fastText	I'm telling you there's a lot of implications to evolution.
src	3) .ابتلاع السيف هو من عادات الهند القديمة
src-Buckwalter	AbtlAE Alsyf hw mn EAdAt Alhnd Alqdymp.
ref	Sword swallowing is from ancient India.
word2vec	The sword is a tradition of ancient India.
morph	The sword of the sword is a traditional Indian tradition.
fastText	Swallow the ball is the old Indian habits.

Table 5: Examples of sentences where word embeddings considering subword information are beneficial.

morphology is a useful signal to incorporate.

It is interesting that the word embeddings that perform best on a word similarity task (morph) do not line up with what performs best in an NMT system (fastText). This reinforces the argument that word similarity tasks alone are not enough to evaluate word embeddings (Faruqui et al., 2016), and that which embeddings we prefer may depend on the downstream task and the dataset. We discuss here briefly the potential strengths and weaknesses of each approach to morphological word embeddings, though more conclusive analysis is left to future work.

One possible reason for the difference in best embeddings between the two tasks, is how in-domain the morphological analyzer is for each task. In the word similarity task, 434 of the 444 unique words in the task receive lemmas (about 98%). On the other hand, in the MT test data, 7,266 out of 8,309 unique words receive lemmas (only about 87%).

It is also possible that function words matter more in the MT task, and that their translation does not improve as much with embeddings informed by lemmas. fastText may help more with these words, especially when function words in English correspond to pieces of a word in Arabic.

From these experiments, it appears that if one is more concerned with semantic similarity or has a dataset that lines up well with the morphological analyzer used to produce lemmas, morphological word embeddings exploiting the morphological resources might be best. On the other hand, for a downstream task such as MT, and when there is a substantial number of words not covered by the analyzer, a method considering character n-grams may be better.

In both cases, word embeddings considering subword information consistently perform better than standard word embeddings on a morphologically rich language such as Arabic. It is possible that future gains could be made by combining the strengths of both models.

7 Conclusion

We extend the skipgram model for word embeddings to incorporate lemmas from a morphological resource in a simple way, maintaining the efficiency of word2vec, and release the code publicly. We show that this model outperforms word2vec and fastText on a word similarity task in Arabic.

We also conduct experiments with these word embeddings as initialization for a low-resource neural machine translation system. We find that the word embeddings utilizing subword information consistently outperform standard word embeddings at this task, and that any of the word embeddings we tried outperformed a random initialization or BPE. fastText does best at this task, with a 0.7

BLEU gain over standard word embeddings and 2.5 BLEU gain over random initialization.

Future work will attempt to combine the strengths of these multiple approaches to incorporating morphological information in word embeddings, as well as to explore other sources of information such as part-of-speech or syntax.

References

Amjad Almahairi, Kyunghyun Cho, Nizar Habash, and Aaron Courville. 2016. First result on arabic neural machine translation. *arXiv preprint arXiv:1606.02680* .

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 349–359. http://aclweb.org/anthology/D15-1041.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 861–872. https://doi.org/10.18653/v1/P17-1080.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics* 5:135–146. http://www.aclweb.org/anthology/Q17-1010.

Jan Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modeling. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Tim Buckwalter. 2002. Arabic transliteration. *URL http://www. qamus. org/transliteration. htm* .

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*. Trento, Italy, pages 261–268.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/view/11000.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 176–181. http://www.aclweb.org/anthology/P11-2031.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 357–361. https://doi.org/10.18653/v1/P16-2058.

Ryan Cotterell and Kevin Duh. 2017. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 91–96. http://www.aclweb.org/anthology/I17-2016.

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*. Association for Computational Linguistics, pages 1287–1292. https://doi.org/10.3115/v1/N15-1140.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1651–1660. https://doi.org/10.18653/v1/P16-1156.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)* 4(1):3.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and improving morphological learning in the neural machine translation decoder. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. volume 1, pages 142–151.

Mattia A Di Gangi and Federico Marcello. 2017. Can monolingual embeddings improve neural machine translation? .

Cicero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character

embeddings. In *Proceedings of the Fifth Named Entity Workshop*. Association for Computational Linguistics, Beijing, China, pages 25–33. http://www.aclweb.org/anthology/W15-3904.

Cicero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *ICML*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. https://doi.org/10.18653/v1/W16-2506.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*. ACM, pages 406–414.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 866–875. https://doi.org/10.18653/v1/N16-1101.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1192–1201. http://www.aclweb.org/anthology/D09-1124.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*. pages 820–828.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel corpus. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 70–78. http://www.aclweb.org/anthology/W17-5704.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*. pages 2741–2749.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, pages 67–72. http://www.aclweb.org/anthology/P17-4012.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, pages 177–180. http://www.aclweb.org/anthology/P07-2045.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, Vancouver, pages 28–39. http://www.aclweb.org/anthology/W17-3204.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1517–1526. http://www.aclweb.org/anthology/P13-1149.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics* 5:365–378. https://transacl.org/ojs/index.php/tacl/article/view/1051.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015a. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1520–1530. http://aclweb.org/anthology/D15-1176.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015b. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586* .

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1412–1421. https://doi.org/10.18653/v1/D15-1166.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. http://www.aclweb.org/anthology/W13-3512.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*. volume 27, pages 466–467.

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.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*. ACM, pages 641–648.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 296–301. http://www.aclweb.org/anthology/I17-2050.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA). http://www.aclweb.org/anthology/L14-1479.

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)*. Association for Computational Linguistics, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 141–150. http://www.aclweb.org/anthology/C14-1015.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, Ahmed Abdelali, Yonatan Belinkov, and Stephan Vogel. 2017. Challenging language-dependent segmentation for arabic: An application to machine translation and part-of-speech tagging. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 601–607. https://doi.org/10.18653/v1/P17-2095.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 86–96. https://doi.org/10.18653/v1/P16-1009.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1715–1725. https://doi.org/10.18653/v1/P16-1162.

Henning Sperr, Jan Niehues, and Alex Waibel. 2013. Letter n-gram-based input encoding for continuous space language models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Sofia, Bulgaria, pages 30–39. http://www.aclweb.org/anthology/W13-3204.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1504–1515. https://aclweb.org/anthology/D16-1157.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1568–1575. https://doi.org/10.18653/v1/D16-1163.