Minimally-Constrained Multilingual Embeddings via Artificial Code-Switching

Michael Wick
Oracle Labs
michael.wick@oracle.com

Pallika Kanani
Oracle Labs
pallika.kanani@oracle.com

Adam Pocock
Oracle Labs
adam.pocock@oracle.com

Abstract

We present a method that consumes a large corpus of multilingual text and produces a single, unified word embedding in which the word vectors generalize across languages. In contrast to current approaches that require language identification, our method is agnostic about the languages with which the documents in the corpus are expressed, and does not rely on parallel corpora to constrain the spaces. Instead we utilize a small set of human provided word translations—which are often freely and readily available. We can encode such word translations as hard constraints in the model's objective functions; however, we find that we can more naturally constrain the space by allowing words in one language to borrow distributional statistics from context words in another language. We achieve this via a process we term *artificial code-switching*. As the name suggests, we induce code-switching so that words across multiple languages appear in contexts together. Not only do embedding models trained on code-switched data learn common cross-lingual structure, the common structure allows an NLP model trained in a source language to generalize to multiple target languages (achieving up to 80% of the accuracy of models trained with target-language data).

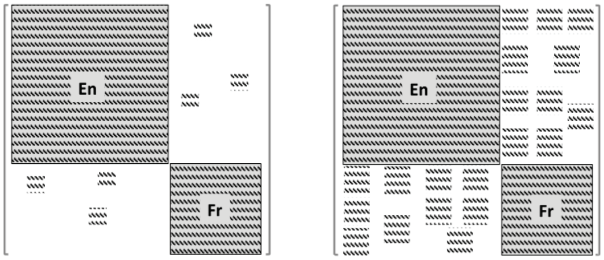
Introduction

An important practical problem in natural language processing (NLP) is to make NLP tools (e.g., named entity recognition, parsers, sentiment analysis) available in every language. Many of the resources available in a language such as English are not available in languages with fewer speakers. One solution is to collect training data in every language for every task for every domain, but such data collection is expensive and time consuming. A second, more feasible solution, is to use large collections of unlabeled multilingual data to find a common representation in which structure is shared across languages. Under such representations, we can train an NLP model in a language with many resources and generalize that model to work on lower resource languages. Thus, such multilingual word embeddings have the potential to substantially reduce the cost and effort required in developing cross-lingual NLP tools.

Word embeddings map word-types to dense, low dimensional (e.g., 300) vectors (Deerwester et al. 1990; Brown, deSouza, and Mercer 1992; Bengio et al. 2003; Mikolov et al. 2013), and are advantageous for NLP because they help cope with the sparsity problems associated with text. Using embeddings learned from monolingual text as features improves the accuracy of existing NLP models (Turian, Ratinov, and Bengio 2010).

The basis for learning such embeddings is the distributional hypothesis of language (Harris 1954; Firth 1957), which stipulates that words are defined by their usage. Learning vector representations of words that are good at predicting their context words over intruder words (Mikolov et al. 2013) captures a remarkable amount of syntactic and semantic structure. For example, vec("king") - vec("man") + vec("woman") $\approx$ vec("queen"). Of course we expect these results are not unique to English. A model trained on French for example would likely yield similar structure: vec("roi") - vec("homme") + vec("femme") $\approx$ vec("reine"). A potential problem with multilingual approaches that depend heavily upon the distributional hypothesis, is that they assume that words are defined by their context in a large corpus of text. While this may hold for words within a single language, it is unlikely to hold across languages because usually all the words in a context belong to the same language. Visualizing the word to context-word co-occurrence statistics as a matrix would reveal large blocks of connectivity for each language, with sparse scattering of non-zero cells elsewhere (see Figure 1a). This block structure causes problems as many word embedding techniques can be seen as performing matrix factorization on co-occurrence matrices (Levy and Goldberg 2014; Pennington, Socher, and Manning 2014; Deerwester et al. 1990).

However, note that some words are shared across languages: *named entities* such as "iPad" or "Obama," *lexical borrowing* of words from other languages (Tsvetkov, Ammar, and Dyer 2015) and *code switching* in which a multilingual speaker switches between languages during a dialogue (Lipski 1978). These phenomena may allow for a jointly trained embedding model to learn structure that generalizes across languages. If we simply combine documents from the different languages under a unified vocabulary and run CBOW on multilingual data, it learns multilingual analogies such as vec("roi") - vec("hombre") + vec("woman") $\approx$ vec("reina"). Thus, we could try to train embeddings on a large enough corpus and see if these phenomena occur with



(a) Bilingal co-oc matrix

(b) ACS co-oc matrix.

Figure 1: A cartoon of a word to context-word co-occurrence matrix for a bilingual corpus (English and French). The matrix is essentially block diagonal, but each block is fairly sparse as it represents the co-occurrence of words within the language. The cartoon on the right shows the matrix after artificial code-switching is applied.

sufficient frequency to learn a good multilingual embedding. Though we expect to do even better if we modify the model or data to more directly capture multilingual structure.

In this paper, we study two ways of improving multilingual embeddings via human provided dictionaries that translate a small subset of vocabulary words across multiple languages. In our initial approach, we augment the underlying embedding method with a set of constraints derived from the word translations. The constraints force dictionary words to have similar magnitudes and angles between them. However, we find that balancing the constraints with the original objective can be challenging. Thus, in our second approach, we transform the data using a process we term artificial code switching (ACS). In much the same way that machine vision practitioners apply affine transformations to their training data to learn invariance to rotation and scale, we apply ACS to our training data to learn invariance to language. The ACS transformation employs the translation dictionaries to replace some of the words in the text with their (possibly rough) translation from another language. Effectively, ACS fills in more cells of the co-occurrence matrix (Figure 1b) making the matrix less block diagonal, and thus ripe for learning multilingual representations.

Our methods improve the quality of multilingual embeddings over a system that relies upon natural cross lingual co-occurrences alone. Further, we provide multilingual word analogy data on which we find that combining multiple languages into a single space enables lower resource languages to benefit from the massive amount of data available in higher resource languages. We determine that ACS in particular learns the best multilingual word embeddings, achieving more than 50% accuracy on bilingual word analogies. Finally, we find that our multilingual embeddings allow us to build sentiment models in languages without training data by training models on English and using the embedding to generalize the information to other languages.

Related Work

Most work on learning multilingual word representations focuses on bilingual data, assumes that the language of each document is known, and requires parallel corpora. At a high level, most approaches employ such aligned translated documents to constrain two monolingual models so that they agree upon a common bilingual representation. For example, by constraining the representations of recurrent neural networks (Sutskever, Vinyals, and Le 2014), multilayer perceptrons (Gao et al. 2014), regularized auto-encoders (Sarath Chandar et al. 2014), multi-task learners (Klementiev, Titov, and Bhattarai 2012), and other models (Hermann and Blunsom 2013) in this way. Unfortunately, in many cases, finding a sufficiently large parallel corpus of paired documents in the source and target languages is difficult, especially for lower resource languages.

Alternatively, canonical correlation analysis (CCA) (Faruqui and Dyer 2014) and deep CCA (Lu et al. 2015) are suitable for mapping word vectors into a common bilingual space. Although these approaches currently employ parallel corpora, they might be able to employ dictionaries instead because they only require alignments across vocabulary words (word-types). However, we do not yet know whether a small handful of word translations is sufficient for CCA, and even so, these approaches would still require language identification when deployed in the wild in order to determine the correct mapping to apply. Furthermore, a potential problem with approaches that apply a global transformation is that the transformation might not be equally appropriate for every word in the vocabulary since it is likely that the monolingual embedding spaces of the two languages are partially (but unevenly) aligned (for example, due to natural phenomena such as lexical borrowing, code-switching and pervasiveness of named entities).

There has also been recent work in relaxing the requirement of aligned parallel corpora. For example, sentence-level alignment is easier to obtain than word level alignment allowing a wider range of corpora to be used. Such sentence-level alignment are naturally captured via auto-encoders (Sarath Chandar et al. 2014) or other models (Hermann and Blunsom 2013) by learning a mapping from the source sentence bag of words (BoW) to a target sentence BoW. Concurrent to our work, Barista is one of the few approaches to eliminate parallel corpora entirely (Gouws and Søgaard 2015). Although justified differently, Barista and artificial code switching both employ dictionaries to transform the data for learning language invariance. Barista is defined for and evaluated in the bilingual setting in which it works remarkably well, allowing an NLP system (a part of speech tagger) trained in one language to generalize to another.

In contrast to the aforementioned approaches which are predominantly bilingual, we combine as many corpora from as many languages as possible into the same embedding space. Note that in many cases, moving from the bilingual to multilingual setting is difficult because the number of constraints scales quadratically with the number of languages. Combining many languages together is not only more convenient (than managing many bilingual pairs), but has the potential to learn better quality embeddings. For example, on a document similarity task, recent work demonstrates that combining similarity scores across four languages results in a more accurate similarity measure than those based on a single language (Hassan, Banea, and Mihalcea 2012).

Multilingual Embeddings

A key motivation for this work is to make progress towards the goal of building multilingual NLP systems using a three step process: (1) Train a multilingual embedding on a large, unlabeled corpus of multilingual documents that span the desired languages. (2) Train NLP models on all available training data, using word embeddings as features. (3) Apply the trained model on data from any of the target languages and achieve accuracy comparable to having in-language training data.

Achieving this goal depends on several key hypotheses: (H1) word embeddings provide sufficient information for training accurate NLP models; (H2) monolingual word embeddings in different languages have similar structure; (H3) a small set of constraints is sufficient for learning this shared structure (H4) the shared structure is sufficient for generalizing NLP models from one language to another.

We expect the extent to which these hypotheses hold depends heavily on the model and task. Further, we expect that the hypotheses themselves are a simplification as we cannot expect simple embedding models to fully capture the richness of language. Three areas which will cause problems with our approach are: linguistic complexities (e.g., varying levels of inflection across languages), word sense disambiguation across languages (e.g., "car" has different meanings in English and Spanish), and cultural diversity (e.g., Spanish has different words for "aunt" depending on whether it is the maternal or paternal aunt). Despite these challenges, in the experiments section we provide initial evidence in support of these hypotheses.

Problem Setting

Suppose we have M languages L_m with corresponding vocabularies V_m, then $V = \cup_{m=1}^{M} V_m$ is the vocabulary of all the languages. We have a large corpus of multilingual text $\mathcal{D}$ with documents $D_i \in \mathcal{D}$ comprised of word sequences $w_1, \cdots, w_{n_i}$ where each $w_j \in V$. We also have a small set of human-provided concept dictionaries $\mathcal{C}$ that link words across languages. A concept dictionary $\mathcal{C}$ is a set of concepts where each concept $C_i \in \mathcal{C}$ is a set of words that all have similar meaning (e.g., a concept set containing "red", "rouge" and "rojo"). Note that we do not necessarily know the language for any given word or document.

Our task is to learn an embedding model $\mathcal{M} : V \rightarrow \mathcal{R}^k$ that maps each word type to a k-dimensional vector such that the vectors capture syntactic and semantic relations between words in a way that generalizes across the languages. We investigate a solution space that is modular in the sense that the multilingual approaches are compatible with many underlying (monolingual) embedding methods. In this way, it is easy to implement the techniques on top of existing embedding implementations such as LSA, RBMs, CBOW, SkipGram, GloVe, or LDA.

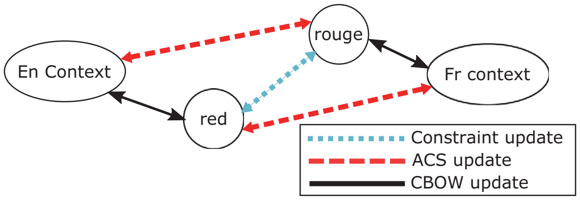


Figure 2: Different updates on word-context pairs.

We begin with a brief description of the area of word embeddings to introduce the necessary terms. Let W be the weights of some underlying embedding model, usually consisting of a matrix in which each row is a word vector. Suppose the underlying embedding model is trained using the following objective function:

$$\mathcal{M} = \text{argmax}_W f(\mathcal{D}; W) \tag{1}$$

For example, the SkipGram/CBOW objective function (Levy and Goldberg 2014), with negative sampling, is approximately: $f(\mathcal{D}; W) = \sum_{\langle w,c \rangle \in \mathcal{D}} \sigma(W_x^T V_c) - \sum_{\langle w,c \rangle \in \hat{\mathcal{D}}} \sigma(W_w^T V_c)$ where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic function, $\hat{\mathcal{D}}$ is the negative dataset and V is the *output* weights. In SkipGram the context is a single word, whereas in CBOW it is the average vector over the context window. Figure 2 shows different kinds of updates that can be performed on a multilingual set of words and contexts. We show the CBOW update as an unbroken black arrow. The update performed by CBOW moves the context closer to the word, and the word closer to the context.

We now present two approaches for learning multilingual embeddings. In the first approach we directly encode the dictionaries as constraints in the underlying word embedding objective function. In the second approach, we use the dictionaries to transform the data in a way that induces code switching between all the languages.

Constraint-based Multilingual Embeddings

We modify the original word embedding objective function (Equation 1) with a constraint term $g(\mathcal{C}; W)$ that encourages the satisfaction of the cross-lingual constraints:

$$f'(\mathcal{D}, \mathcal{C}; W) = f(\mathcal{D}; W) + g(\mathcal{C}; W) \tag{2}$$

There are many possible ways of defining g, but defining it in a way that balances the dictionary concepts with the data is surprisingly difficult. One possibility is to represent each concept set as a vector and then update words to increase their probability under both their context-word and concept-set vectors. However, we find that this approach does not work in practice because the constraints imposed by the data (context) dominate the constraints imposed by the concept sets. Instead, we employ a more aggressive form of constraints that encourage words in the same concept set to have small angles between and similar magnitudes to each other:

$$g(\mathcal{C}; W) = -\sum_{c \in \mathcal{C}} \sum_{w_i \neq w_j \in c} \sigma(\cos(W_i, W_j)) \tag{3}$$

where $\mathcal{C}(w) = \{c \mid c \in \mathcal{C} \wedge w \in c\}$ are concepts containing w. In Figure 2 the constraint updates are shown as a blue dotted line, representing the constraints pulling the vectors for "rouge" and "red" closer together. As these two words are pulled together the context words also move closer.

Artificial Code-Switching

Code-switching is the process in which a speaker of multiple languages switches between those languages in discourse. Consider the utterance "*pizza khaneka mood nahi*" which translates to "not in the mood to eat pizza." The Hindi verb "khaneka" which means "to eat" and the English noun "pizza" are able to share distributional information via the code-switched utterance. As we can see, code-switching allows the distributional meaning of a word in one language to borrow from context in another language, thus providing a rich base from which to learn window-based embeddings.

Unfortunately, in written text—and especially concerning certain European languages such as French—code-switching is an infrequent event. Thus, one approach would be to use a classifier to identify instances of code-switching and treat such contexts in a special way (e.g., by giving higher weight to the updates from the code-switched data). However, as illustrated by the example utterance (which contains two transliterated Hindi words and two English words), the problem of language identification is non-trivial, especially for social media (e.g., tweets). Also, it is not clear that sufficient natural code-switching occurs in large datasets such as Wikipedia.

Instead, we use our dictionaries to artificially induce extra code-switching in the input data. This process, which we term artificial code-switching (ACS), fills in unobserved cells in the word to context-word co-occurrence matrix (see Figure 1b). This extra knowledge is analogous to having extra recommendations in a recommender system (i.e., recommendations that a word in one language could be substituted for a word in another language). An interesting question is how to fill the cells of this matrix in a way that most naturally causes the learning of shared structure in the multilingual space. Ideally, in order to respect the distributional hypothesis, we want the co-occurrence statistics between words of different languages to resemble the co-occurrence statistics of words within a language, while at the same time, preserving the co-occurence statistics of words within a language. A simple way of accomplishing this is to fill in the matrix by randomly replacing a word in one language with its translation in another. In this way, co-occurrence "mass" from the monolingual blocks is shared across languages.

Specifically, we generate a new code-switched corpus $\mathcal{D}'$ by transforming each word $w_i \in \mathcal{D}$ into a new word w'_i with a transformation distribution $q(w'_i|w_i)$ defined as follows. First, we draw a variable $s \sim Bernoulli(\alpha)$. If $s = true$ then we code-switch and sample a new word w'_i. To generate w'_i we sample a concept c from $C(w_i)$ then we sample a word w'_i from c. If $s = false$ then we do not code switch.

Looking at Figure 2 again, we can see that the code-switching update moves the English word "red" closer to the French context for the word "rouge" and vice versa. This does not directly affect the relationship between "red" and "rouge" but over repeated updates it enforces a relaxed form of the constraint update given in the previous section.

Of course, more complex models of code-switching are possible, but our goal is not to model the phenomena of code-switching; rather, we are interested in exploiting it to introduce multilingual structure in our embedding space.

Remark: we can more precisely characterize the underlying objective function of ACS with SkipGram. To briefly sketch this, note that SkipGram can be seen as factorizing the shifted word to context pointwise mutual information (PMI) matrix, which depends only upon unigram and word-pair probabilities. Thus, we can combine our definition for $q(w'|w)$ with the unigram and word-pair probabilities of the original data $\mathcal{D}$ to express the corresponding quantities for the ACS data $\mathcal{D}'$ from which we can express the cells of the matrix ACS produces. We then compare matrix cells corresponding to word-pairs within a language with word-pairs across languages, and see that code-switching parameter α balances how mass is distributed across these two cell types.

Experiments

The purpose of our experiments is to assess the quality and utility of the multilingual embedding spaces. The first set of experiments measures the former, and the second set measure the latter on the task of sentiment analysis. We select five languages to represent various levels of resource-availability, as reflected by the number of Wikipedia pages. English has almost five million pages in Wikipedia, French, German and Spanish each have over a million, whereas, Bokmål has over 100,000 articles. See Table 1 for a list of languages. We also considered languages with even fewer Wikipedia pages, but found a large proportion of the pages to be stubs, and hence less useful.

Systems

Each system employs CBOW as the underlying model f. The multilingual embedding models also employ a set of human provided concept dictionaries $\mathcal{C}$ that translate words with similar meaning from one language to another. Such dictionaries are readily available, and for the purpose of these experiments we use OmegaWiki,[1] a community based effort to provide definitions and translations for every language in the world. Systems include:

- **Monolingual** - A baseline system in which separate embeddings are trained for each language on monolingual data with CBOW.
- **CBOW No constraints (no const)** - A baseline system in which we train CBOW on a multilingual corpus.
- **CBOW With constraints (with const)** - The method described by setting g to Equation 3. After each CBOW update, we perform updates to satisfy g on words in the context for which we have a constraint.
- **Artificial code switching (ACS)** - The artificial code switching approach in which we use the concept sets in OmegaWiki to perform the word substitutions. We set the parameter $\alpha = 0.25$.[2]

The data in our experiments comprises interleaved documents from the various mono-lingual Wikipedias: bilingual experiments involve Wikipedia documents in two languages; multilingual experiments use Wikipedia documents

[1] http://www.omegawiki.org

[2] $\alpha = 0.25$ is our initial guess for the parameter; we find on development data that the method is robust to the setting of α.

	Embedding Data		Sentiment Data			Target Language Baseline	
Language	#Docs (x10^6)	#Concepts	#Train	#Test	%Tw	TwA%	TotA%
English (en)	4.87	8821	24960	6393	36.0%	63.8%	68.4%
French (fr)	1.62	7408	14284	3081	36.0	69.9	74.9
German (de)	1.82	8258	12247	2596	25.5	70.2	74.9
Spanish (es)	1.18	6501	16506	59529	84.7	64.4	64.2
Bokmål (no)	0.41	5336	1225	1000	0.00	-	72.9

Table 1: Datasets. Sentiment accuracy on just twitter documents (TwA) and all documents (TotA).

Method	**Word-pair cos.**	**En W. Analogy**	**Fr W. Analogy**	**Mixed En+Fr W. Analogy**
no const	0.286	77.5%	47.8%	39.3%
with const	0.422	52.8	40.4	43.6
ACS	0.439	66.9	53.3	52.6
Monolingual	-0.015	81.1	45.2	NA

Table 2: Comparison of multilingual embeddings.

from all five languages. In all experiments, we use the same CBOW parameters (2 iterations, 300 dimensions, learning rate 0.05, filter words occurring fewer than 10 times).

Evaluation of embedding space

In this experiment we evaluate the quality of the joint multilingual embedding space. We assess two aspects of the spaces. First, the amount of structure shared across languages. Second, as the number of languages increases in the shared space, how does the quality of the individual language's representation change in this space.

For the latter aspect, we examine the accuracy of multilingual embeddings on the word analogy task (Mikolov et al. 2013). An example analogy question is *man:king::woman:?*. The embedding makes a prediction via $king - man + woman$ and if (and only if) $queen$ is one of the top five most similar words to the resulting vector then the model is credited with getting the analogy correct. The original dataset contains analogies in English, but we translate them into French.

For evaluating the former aspect, first, we create a mixed bilingual (En+Fr) analogy tasks by mixing words from the monolingual analogies, e.g., *homme:roi::woman:queen*. Second, we split the OmegaWiki concepts into 50/50 training/testing sets. We train the embedding models using half the concepts, and use the other half for evaluating the quality of the embedding (via the average cosine similarity of words that appear in the same concept).

We present the evaluations in Table 2. All models are trained on the combination of English, French, Spanish, German, and Bokmål, but the monolingual models are trained on each language independently. First, observe that CBOW alone (with no constraints) learns a considerable amount of cross-lingual word similarity (0.286 cosine similarity). This is likely due to the phenomena mentioned in the introduction (e.g., lexical borrowing), and is notable because multilingual word-vectors from independently trained embeddings should be orthogonal (as verified by last row of the table). Both multilingual embedding techniques substantially improve the cosine similarity over multilingual CBOW, with ACS achieving the most improvement. Furthermore, ACS performs substantially better on the mixed (En+Fr) word analogies than the other systems. The results indicate that ACS learns common multilingual structure.

We also measure accuracy on English and French monolingual word analogies. On English, a high resource language, the multilingual information hurts performance. However, for French—which has an order of magnitude fewer tokens than English—the multilingual information improves the quality of the embedding, especially with ACS, which again performs best. We believe that the reduction of English performance is because each English word is trained less in the ACS system, thus we expect more training iterations will compensate for the drop in English performance. We conclude that the shared structure allows a low resource language to benefit from a high resource language (though, at some cost to the high resource language when using the same number of iterations).

Together, these results provide some evidence for the hypotheses and assumptions stated earlier. In particular, that a substantial amount of structure is shared across languages, and that artificial code switching is able to capture such structure. One reason ACS is so effective is that it is able to overrule concept-based constraints when they contradict meaning derived naturally from the data.

Evaluation on sentiment

The purpose of this experiment is to test if our multilingual embeddings allow us to train an NLP model on a high resource language (English), and then evaluate it on languages for which no training data exists. We do not expect high performance, but instead hope to investigate how close such an approach gets to models trained on in-language data. We evaluate on document-level sentiment analysis, the task of classifying a document as expressing overall positive, negative or neutral sentiment. Thus, the success of multilingual embeddings hinges crucially upon whether the sentiment information captured in the word embedding dimensions generalizes across languages. We have labeled sentiment data (three classes) in the five languages: English, Spanish, French, German and Bokmål. The data comprises various sources such as product reviews, social media streams, and the micro-blogging site, Twitter (see Table 1). We supplement our own datasets with additional Spanish (Hu and Liu 2004) and English data (Nakov et al. 2013).

In order to establish in-language baselines for the target languages (last two columns of Table 1), we train sentiment classifiers on each of the target language's training data, using unigrams, bigrams, and bias, but no embeddings as features (termed *target language baselines*). The ultimate goal is to achieve accuracy similar to these systems, but without in-language training data. To control for the amount of natural lexical overlap and code-switching, we establish cross-lingual baselines by training the lexical-based classifier on English. Additionally, we train the cross-lingual models on English using only the multilingual (all five languages) word embeddings as features (i.e., no lexical and bias features) by averaging the normalized word vectors in each document. Finally, we train the cross-lingual embeddings on French to evaluate English as a target language.[3] We use FACTORIE for training (McCallum, Schultz, and Singh 2009).

We present the results in Table 3. The rows are grouped according to the test language. For each group, the target language baseline can be found in Table 1. We report the percentage of the target language baseline's accuracy that each system achieves. In other words, we ask, 'what fraction of accuracy is obtained by using only source training data instead of target language training data.' The two columns under the bilingual and multilingual sections of the table represent fraction of the target language baseline accuracy for Twitter "TwA%" and complete datasets respectively ("TotA%"). In many cases, source-trained models achieve a high fraction of the accuracy of the *target language baseline* without using target training data. On Twitter data, the fraction is especially high, often over 80%. This is due to the fact that the short tweets have a high information content per word, and sentiment bearing words contribute more weight to embedding-dimension features. Finally, we report the average performance (over all five languages) in Figure 3.

Overall, the sentiment results provide some evidence for answering the hypotheses outlined at the beginning of the manuscript: that the multilingual embeddings can be used for cross-lingual NLP. However, there is still much progress to be made before such cross-lingual systems could actually be deployed in the real-world. Possible ways of improving the system include (1) task-specific supervised fine tuning which would produce better embeddings for sentiment-bearing terms (Collobert et al. 2011), (2) better ways of incorporating embeddings as features such as RNNs (Socher et al. 2013) or via non-linear classifiers, and (3) alternative ways of incorporating concepts as constraints.

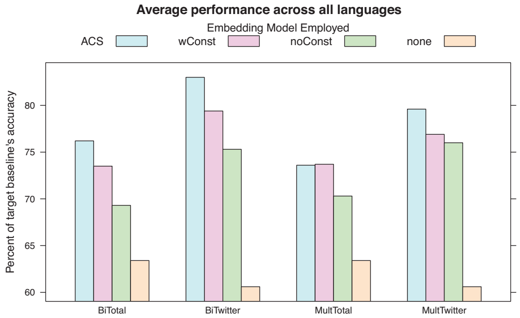


Figure 3: Average cross-lingual sentiment performance.

		Bilingual		Multilingual	
Tr-Te	Model	TwA%	TotA%	TwA%	TotA%
en-fr	None	53.1	57.3	53.1	57.3
en-fr	No Const	77.0	64.6	79.0	67.8
en-fr	W/ Const	73.1	65.9	78.4	**71.4**
en-fr	ACS	**84.0**	**67.4**	**80.5**	65.1
en-es	None	59.3	60.4	59.3	60.4
en-es	No Const	65.2	75.4	**77.4**	69.1
en-es	W/ Const	**81.5**	79.0	71.4	67.8
en-es	ACS	78.3	**85.5**	76.5	**78.4**
en-de	None	68.5	63.3	68.5	63.3
en-de	No Const	77.9	58.5	**80.9**	62.7
en-de	W/ Const	80.5	64.2	76.2	**68.8**
en-de	ACS	**85.4**	**74.7**	79.5	66.7
en-no	None	-	87.9	-	87.9
en-no	No Const	-	89.6	-	**94.4**
en-no	W/ Const	-	**93.6**	-	89.0
en-no	ACS	-	89.6	-	88.8
fr-en	None	61.6	48.2	61.6	48.2
fr-en	No Const	81.0	58.2	66.5	57.6
fr-en	W/ Const	82.5	**64.7**	**81.6**	**71.7**
fr-en	ACS	**84.3**	64.0	**81.6**	69.2

Table 3: Sentiment across language: Tr=training; Te=testing.

Conclusion

We have presented two methods for training multilingual word embedding models that integrate multiple languages into a shared vector space. These methods require only a small dictionary of translated words to align the vector spaces, allowing useful inferences to be made across languages, based solely upon the vectors. The constraint approach introduces links between the words themselves, and the artificial code switching gives a softer link between a word and a context in a different language. Both of these approaches allow us to generalize a model trained on one language, and recover much of the test performance in another language. We also demonstrated that an embedding model can contain more than two languages, despite the problem of multilingual polysemy. These results are encouraging as we used simple techniques for incorporating the word embeddings into classification tasks, and there exist many exciting approaches for incorporating embeddings into monolingual NLP tasks which we could apply to the multilingual setting with this work. There are several other directions we wish to investigate to improve the performance of this system, focusing on how to extract the most benefit from the scarce translation resources available in many languages.

[3] We also trained on Spanish, German and Bokmål, but the results were similar and we omit them for space.

Acknowledgements

We thank Randall Sparks and Don Hindle for providing multilingual sentiment data and feedback, our Oracle Labs team members for discussion, and the reviewers for suggestions.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research* 3:1137–1155.

Brown, P. F.; deSouza, P. V.; and Mercer, R. L. 1992. Class-based n-gram models of natural language. In *ACL*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12:2493–2537.

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

Faruqui, M., and Dyer, C. 2014. Improving vector space word representations using multilingual correlation. In *EACL*.

Firth, J. 1957. Synopsis of linguistic theory. *Studies in Linguistic Analysis*.

Gao, J.; He, X.; Yih, W.; and Deng, L. 2014. Learning continuous phrase representations for translation modeling. In *Proceedings of Association for Computational Linguistics*.

Gouws, S., and Søgaard, A. 2015. Simple task-specific bilingual word embeddings. In *NAACL*, 1386–1390. Denver, Colorado: Association for Computational Linguistics.

Harris, Z. 1954. Distributional structure. *Word*.

Hassan, S.; Banea, C.; and Mihalcea, R. 2012. Measuring semantic relatedness using multilingual representations. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, 20–29. Association for Computational Linguistics.

Hermann, K. M., and Blunsom, P. 2013. Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. ACM.

Klementiev, A.; Titov, I.; and Bhattarai, B. 2012. Inducing crosslingual distributed representations of words. In *COLING*, 1459–1474.

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

Lipski, J. 1978. Code-switching and the problem of bilingual competence. *Aspects of bilingualism* 250:264.

Lu, A.; Wang, W.; Bansal, M.; Gimpel, K.; and Livescu, K. 2015. Deep multilingual correlation for improved word embeddings. In *NAACL*.

McCallum, A.; Schultz, K.; and Singh, S. 2009. FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems (NIPS)*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *ICLR 2013, Workshop track*.

Nakov, P.; Zornitsa, K.; Alan, R.; Sara, R.; Veseline, S.; and Theresa, W. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. *Atlanta, Georgia, USA* 312.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12.

Sarath Chandar, A.; Lauly, S.; Larochelle, H.; Khapra, M.; Ravindran, B.; Raykar, V.; and Saha, A. 2014. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, 1853–1861.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Sutskever, I.; Vinyals, O.; and Le, Q. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 3104–3112.

Tsvetkov, Y.; Ammar, W.; and Dyer, C. 2015. Constraint-based models of lexical borrowing. In *NAACL*.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*.