Multilingual Word Embedding for Zero-Shot Text Classification

YAOYAO DAI[*]
New York University Abu Dhabi
BENJAMIN J. RADFORD[†]
University of North Carolina at Charlotte

ABSTRACT

For years political scientists have been developing tools to analyze text data, motivated by both the richness and wide availability of unstructured text as well as by the limited availability of accurate structured data. However, while many research questions are comparative and cross-national, we lack methods for analyzing multilingual corpora. Political scientists typically analyze texts from multilingual corpora separately and within the contexts of each individual language or by translating all texts into a common language before performing analysis. In this paper, we develop a Zero-shot Bilingual Classifier (*0-BlinC*), a novel multitask feed-forward neural network that utilizes cross-lingual information to facilitate text classification in multilingual corpora. Using a parallel bilingual corpus and training data in a single source language, *0-BlinC* can perform quasi-sentence-level text classification in a target language for which no training labels are available. We demonstrate our method by measuring policy positions of party manifestos in English, Spanish, Bulgarian, Estonian, Italian, German and French using labeled text in English only. *0-BlinC* is shown to outperform alternative methods that include the use of a machine translation service and pre-trained word vectors.

[*] Post-doctoral Associate, Social Science Division, NYU Abu Dhabi. (yaoyao.dai@nyu.edu).

[†] Assistant Professor, Department of Political Science and Public Administration, University of North Carolina at Charlotte.(benjamin.radford@gmail.com).

1 Introduction

Social scientists increasingly recognize the importance and usefulness of analyzing text as data. While the conventional structured data produced by governments and other organizations are often unavailable, misleading, and manipulated, countries around the world are producing large amount of text data daily, at both the government level and individual level. Using massive social media corpora, news articles, and public speeches, scholars are able to study various aspects of the black-box of authoritarian politics such as censorship, media control, and bureaucratic relationships (King, Pan and Roberts, 2017; Pan and Chen, 2018). Using the language features of political texts, scholars have measured properties such as political sophistication in communication (Benoit, Munger and Spirling, 2017), belligerent international statements (Schrodt, 2000), and campaign rhetoric (Crabtree et al., 2018). In addition to language features, advances in text analysis also enable us to measure theoretical concepts such as policy postion, ideology, populism, and polarization (Lowe et al., 2011; Slapin and Proksch, 2008; Hawkins and Silva, 2018; Peterson and Spirling, 2018), and to automatically identify events of interests such as different types of conflicts (Boschee et al., 2015).

However, most advances in text analysis in political science apply only to single language corpora. This is problematic since many political science questions are comparative, cross-national, and cross-lingual in nature. There are two common approaches in dealing with multilingual corpora. Scholars either translate all documents into the same language or analyze documents in different languages individually (Lucas et al., 2015). Both approaches suffer from high cost and low efficiency. In the first approach, scholars often use machine translation (MT) services, such as Google Translate, as a cheaper and faster alternative to human translation. However, machine translation services are not free and generally limit the size of translation requests. Google Cloud Translation charges $20 per 1,000,000 characters, including white space. Moreover, those cloud translation services are optimized for short requests. Google Cloud Translation recommends each translation request be shorter than 2000 characters and rejects large requests.[1] For example, a typical political manifesto, a common text for the study of political discourse, is about 40 pages long with over 160,000 characters. This means that we would need to make at least 80 requests to translate this single document. To translate a medium-sized corpus of 2000 manifestos would cost over $6,000. For larger corpora such as news archives, the costs may become prohibitive for an individual researcher or research

[1] The quotas and price can be found at https://cloud.google.com/translate/quotas.

teams with limited resources.

The second approach is often used in applications of supervised learning and the dictionary method. Scholars first hand-code a sample of documents or use pre-existing labelled documents in each language; then they build a separate classifier in each language. Depending on the languages of interests, it could be very expensive to find human coders to code even a small sample of data. Moreover, The knowledge learned in one language is irrelevant to and discarded in the modeling in an additional language. It seems inefficient to train individual models in each language. In reality, texts in some languages have been studied more extensively than others. We have larger and better labelled texts in some languages but not others. The size of pre-existing labelled data in a certain language of interest may be too small to train a model that performs well.

In this paper we introduce 0-BlinC (Zero-shot Bilingual Classifier), a novel multitask feed-forward neural network to create multilingual word embeddings that enable both supervised and unsupervised learning using multilingual parallel corpora without translation. We apply our method to a supervised classification task where we measure policy positions using party manifestos in English, Spanish, Bulgarian, Estonian, Italian, German and French. We demonstrate that it is easy to incorporate the classification model into our multilingual neural network framework. Our method jointly optimizes the classification and multilingual word embedding models, allowing knowledge learned in one model to simultaneously inform the training of the other model. Using this method, we are able to fit a model using labelled English manifestos only and to accurately measure policy positions in Spanish, Bulgarian, Estonian, Italian, German and French manifestos, where no training data exists. We further demonstrate that this method outperforms the MT approach described above as well as a method that uses pre-trained (PT) bilingual word vectors rather than a multitask model.

2 Method

Our 0-BlinC model is a multitask feed-forward neural network (NN). It consists of four sub-models, or tasks: word2vec in language *A*, word2vec in language *B*, word embedding alignment with parallel corpus in languages *A* and *B*, and a text classifier in language *A*. A diagram of the model architecture is given in Figure 1. The model's forward pass is represented from top to bottom; four input nodes at the top of the

Figure 1: Representation of zero-shot text classifier language model.

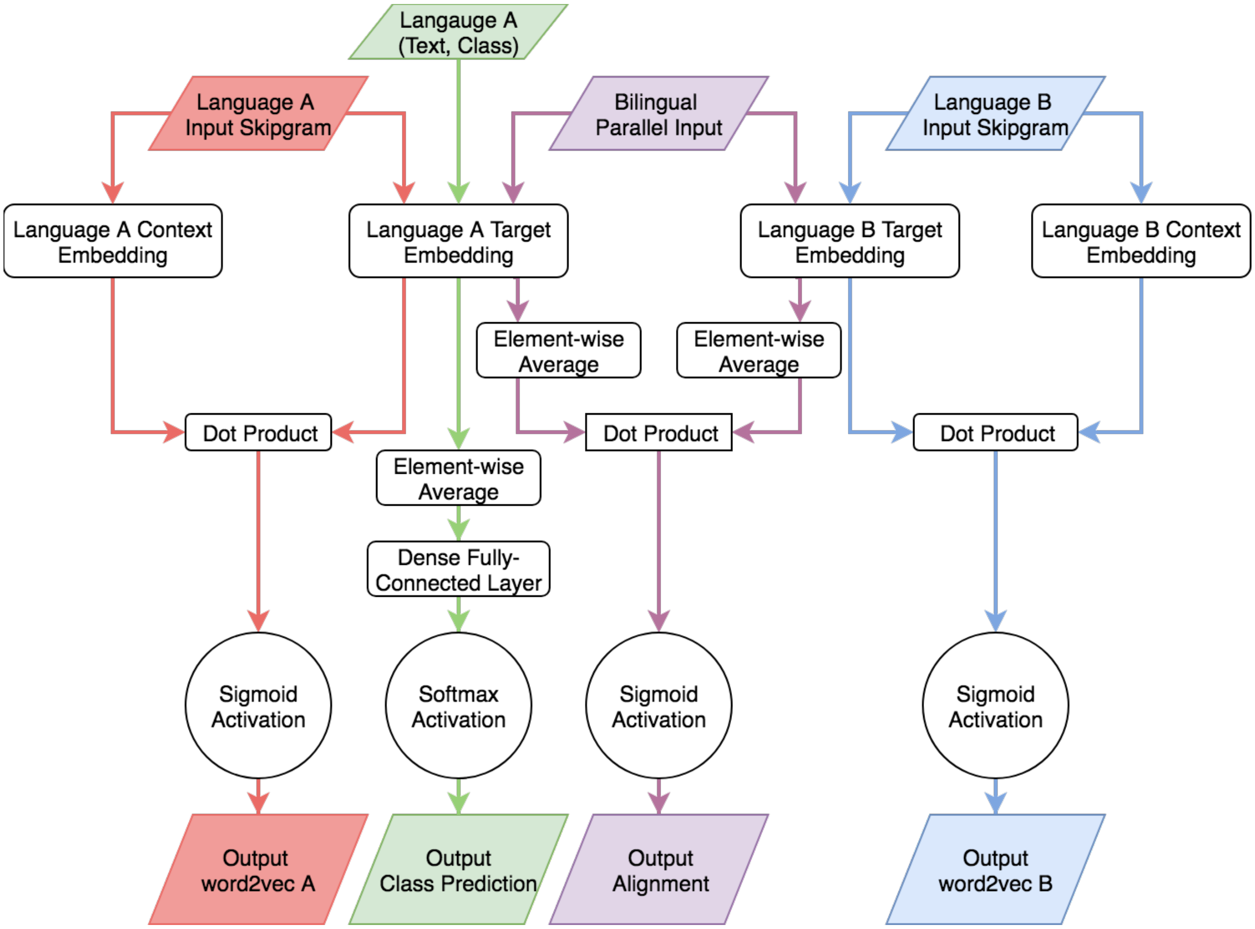


diagram represent the input data for each model task. For clarity, we have omitted all regularization layers from Figure 1.[2]

The four sub-models are simultaneously fit using adaptive moment estimation (Adam), a gradient descent optimizer (Kingma and Ba, 2015), by minimizing loss function given in Equation 1. Equations 1a and 1b are the standard loss functions for skipgram negative sampling word2vec in languages *A* (source) and *B* (target), respectively (Mikolov, Sutskever, Chen, Corrado and Dean, 2013). Equation 1c is a novel cross-lingual negative sampling loss function for aligning word vectors across languages *A* and *B*. Equation 1d is standard multiclass logarithmic loss with M distinct classes and probability p_{ij} assigned to observation i with respect to class j. The overall loss of our model is a weighted sum of the four sub-models with weights indicated by γ.[3] v_ti and v_ci indicate target and context word vectors associated with observation i. $\bar{v}_ti$ indicates the element-wise average of target embedding word vectors for observation i, essentially a "sentence vector." The logistic function, $\frac{1}{1+e^{-x}}$, is denoted by σ. We explain each sub-model and its loss function in more detail in the following sections.

$$\text{Loss} = \gamma^{langA}\left(-\frac{1}{N}\sum_{i=1}^{N} y_i^{langA}\log\left(\sigma\left(v_{ti}^{langA}\cdot v_{ci}^{langA}\right)\right)\right) \tag{1a}$$

$$+\,\gamma^{langB}\left(-\frac{1}{N}\sum_{i=1}^{N} y_i^{langB}\log\left(\sigma\left(v_{ti}^{langB}\cdot v_{ci}^{langB}\right)\right)\right) \tag{1b}$$

$$+\,\gamma^{parallel}\left(-\frac{1}{N}\sum_{i=1}^{N} y_i^{parallel}\log\left(\sigma\left(\bar{v}_{ti}^{parallelA}\cdot \bar{v}_{ti}^{parallelB}\right)\right)\right) \tag{1c}$$

$$+\,\gamma^{classifier}\left(-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}^{classifier}\log\left(p_{ij}^{classifier}\right)\right) \tag{1d}$$

2.1 Word2vec in a Single Language

Unlike the commonly "bag-of-words" representation of words and documents common in political science, 0-BlinC uses featurized representation of words obtained through the word2vec model, a neural network language model (NNLM) . Instead of treating each unique word as one dimension in a vector of length V, the vocabulary size, word2vec represents words in a dense vector space with many fewer dimensions.

[2] We use dropout and add Gaussian noise to layers of the alignment and classification sub-models. This helps to prevent overfitting of the classifier to the source language training set.

[3] We use the following weights, chosen with respect to English out-of-sample performance: $\gamma^{langA} = 1.0$, $\gamma^{langB} = 1.0$, $\gamma^{parallel} = 3.0$, and $\gamma^{classifier} = 0.1$.

Word2vec positions semantically and syntactically similar words, those that share common contexts, close to each other in this vector space.[4] A word's representation within this vector space can be referred to as its "word vector" or "embedding."

Word2vec has recently found widespread adoption in NLP tasks (Mikolov, Chen, Corrado and Dean, 2013). Word2vec encompasses several different, related models including the continuous bag-of-words (CBOW), skipgram (SG), and negative-sampling SG (NSSG). The SG model inputs a target word from a text and attempts to predict the target word's likely context words, words that are co-located with the target. The CBOW model does the reverse: given a set of context words, CBOW attempts to predict the context's target word. The NSSG model is an efficient method for approximating the SG variant of word2vec. We focus on NSSG because the negative-sampling strategy is central to our multitask model.

NSSG word2vec takes as input target word and context word pairs (tuples). Given a window-size, k, tuples are generated by taking the target word at index i and pairing it with all context words from $i - k$ to $i + k$.[5] For every target-context word pair, we also generate s negative samples, target-context word pairs that are not observed in the actual text corpus.[6] The positive samples, co-located pairs that are observed in the corpus, are assigned a value of 1. Negative samples are assigned a value of 0. NSSG word2vec is tasked with discriminating between positive samples and negative samples given the input word pairs. Therefore, NSSG word2vec is a logistic regression model.

Input words to word2vec are represented as one-hot-encoded vectors of length V, where V is the number of unique words in the corpus's vocabulary. A one-hot-encoded vector is a vector of $V - 1$ zeros and a single value of one at the index of the word in question. The input word, represented as a one-hot-encoded vector, is then multiplied by a dense, real-valued weights matrix of size $V \times d$, where d is the chosen size of the hidden layer (i.e. "embedding").[7] By multiplying the $1 \times V$ input vector for a word with the $V \times d$ weights matrix, a $1 \times d$ vector is generated; this is the word's vector representation, v_{word}. Every word has a unique vector representation. The predicted value given an input pair is computed by taking the dot product of the target word vector and the context word vector and then applying the logistic function, $\sigma(\cdot)$. A gradient descent algorithm, typically stochastic gradient descent, is then used to select parameters (weights) that minimize the logarithmic loss between $\sigma(v_{\text{target word}} \cdot v_{\text{context word}})$ and the true value $[0, 1]$.

[4] A target word's context is the set of words immediately surrounding it.

[5] The actual window size is randomly sampled, per target word, from 1 to k. This effectively causes more distant context words to be sampled with less frequency than nearer context words. We select $k = 10$.

[6] We select $s = 15$.

[7] We choose $d = 300$, in keeping with standard practice.

This has the effect of causing positive target-context word vector pairs to be similarly-valued and negative target-context word vector pairs to be dis-similarly-valued.

2.2 Negative Sampling for Alignment

The monolingual word2vec model learns the featurized representation of words, or word vectors, for the individual languages *A* and *B*. The alignment model learns to update the word vectors in language *A* and *B* into the same vector space so that similar words in the two languages are represented close to each other in the same vector space. To achieve this, our model requires a parallel corpus with sentence-aligned text in languages *A* and *B*. We refer to this as the parallel corpus. Our parallel corpora are drawn from the Europarl Parallel Corpus (Europarl), which is derived from the proceedings of the European Parliment (Koehn, 2005). Europarl contains the same documents in 21 European languages with matched meanings at the sentence level. Although we could train the word embeddings for languages *A* and *B* using only the parallel corpus, we recommend training the word2vec sub-models on all available text data. This guarantees that words found in only one corpus are still learned by the model for future classification tasks. Therefore, we include each language from the parallel corpus and the associated monolingual corpus in each monolingual word2vec sub-model. While we do not use the target language *B*'s class labels during the model training stage (they are only used for model evaluation), we *do* include the target language *B*'s texts in the target language *B*'s word2vec sub-model. This is not necessary – the target language *B*'s word2vec model could be trained solely on the texts found in the parallel corpus – but we suspect that it will lead to a better target language embedding model.[8]

Our alignment model uses a similar negative sampling algorithm as in word2vec. It accepts as input a pair of sentences from languages *A* and *B*. A sample of five words is drawn from each sentence according to Zipf's law, such that common words are less likely to be sampled and uncommon words more likely to be sampled (Powers, 1998).[9] For one third of all parallel sentence pairs in our training data, the sentence pair should be drawn from a parallel corpus; that is, the sentences should have roughly the same meaning in both languages. For one third of the remaining parallel sentence pairs, negative samples are drawn by sampling words randomly from language *B* according to Zipf's law and pairing those random words with true sentences from language *A*. For the final one third of the training sentence pairs, negative samples are

[8] We leave the demonstration of this to future work.

[9] We use the `make_sampling_table` function from Keras to generate our probability vector. Words are sampled inversely proportional to Zipf's law with $s = 1$. For more information, see Chollet et al. (2015).

drawn from language *A* and paired with real sentences from language *B*. The word embedding alignment sub-model then attempts to predict whether each input pair is a true pair (1) or a negative-sampled pair (0). This method for bilingual word vector alignment is what Ruder, Vulić and Søgaard (2017) would call joint optimization with sentence-aligned data. The sets of words input into the alignment model are projected into their respective language's target embedding layer, the same embedding layer used for target word embedding in the word2vec sub-models.[10]

2.3 Text Classification in One Language

To perform the classification task, we add a classification sub-model that is fit using only the labelled texts from language *A*. The classification sub-model takes as input a sample of words from a given sentence and outputs a vector of predicted probabilities of class membership. For a given sentence, we sample $s = 5$ words from the sentence according to an inverse Zipf's distribution, where the word frequency estimates are based on their frequency in the full corpus. This has the effect of down-sampling common words and up-sampling uncommon words. We then compute the element-wise average of the target-embedding word vectors for these s words. The result is a single $1 \times d$ "sentence vector" that represents the input sentence. We multiply this sentence vector by an additional weights matrix of size $d \times M$, where M is the number of output classes. The output vector, p, of size $1 \times M$, is then softmax-normalized: $p_j = \frac{e^{p_j}}{\sum_{m=1}^{M} e^{p_m}}$. This sub-model is fit by minimizing the multiclass logarithmic loss given in Equation 1d.

To summarize, our 0-BlinC model learns to produce a shared vector space for language *A* and language *B* through training word2vec in each language and aligning the word vectors via a parallel corpus. The parallel corpus does not need to be the same as the corpus used in training monolingual word2vec models and can be unrelated to the topic of interest. 0-BlinC trains a classifier in language *A*, and language *A* only, that uses the word vectors from language *A*'s word2vec sub-model to predict class membership. The classifier trained in language *A* is then used to classify texts in language *B* for which no labelled training data exists.

[10] Sentences over 20 words in length are omitted from the parallel corpus in order to minimize the chance that sampling from long sentences leads to spurious paired bags of words.

3 Application and Evaluation

We evaluate our method in classifying policy areas in party manifestos in seven different languages. The corpus we use is the Manifesto Corpus from the Manifesto Project (MRG/CMP/MARPOR) (Volkens et al., 2018). Political manifestos are a common choice of political discourse in various social science research, such as partisan ideology, populism, and party families (Rooduijn, de Lange and van der Brug, 2014; Hawkins and Silva, 2018). The Manifesto Corpus of the MP project is particularly good for assessing our model because it is a multilingual corpus with documents in more than 35 languages from over 50 countries and it contains high quality hand-coded policy areas at the quasi-sentence level; these provides us the "ground truth" necessary to evaluate our model.

The MP project codes each quasi-sentence into one of 7 major policy areas and 1 other or residual category. The 7 major policy areas are *External Relations*, *Freedom and Democracy*, *Political System*, *Economy*, *Welfare and Quality of Life*, *Fabric of Society*, and *Social Groups*. Each of the major areas are further split into subcategories that capture specific aspects of each major area. While one sentence might contain several issues, the MP project cuts those sentences into quasi-sentences, each of which only contains one message. Therefore, each quasi-sentence only has one class/policy area. As an evaluation of our model, we use the top level eight categories as the target classes we try to predict.

We perform English-to-Target classification transfer on six language pairs: English-Spanish, English-Bulgarian, English-Estonian, English-Italian, English-German and English-French. The seven languages belong to four different language families: Romanic (French, Italian and Spanish), Germanic (English and German), Slavik (Bulgarian), and Finni-Ugric (Estonian).[11]

We perform the same minimum pre-processing to documents in all 7 languages: we remove white space and common punctuation, and covert all words to lowercase. After pre-processing, we are left with 133,184 quasi-sentences in Spanish, 114,770 in German, 11,136 in Estonian, 10,641 in Bulgarian, 4,173 in Italian, and 22,343 in French. In training the word2vec models, we use all the documents in the language pairs. In training the classifier, we split our English manifesto corpus into a training set (75%) and a test set (25%). The target language labels are completely unobserved during model training. The model is fit by adaptive moment estimation (Adam), a gradient descent optimizer (Kingma and Ba, 2015). We train our

[11] It is common practice in word2vec to drop uncommon terms from the vocabulary. Here, we set a minimum per-word count of 20 occurrences across all corpora for inclusion. Quasi-sentences from the Manifesto Project are only omitted if they contain zero words after infrequent words are dropped for word2vec. Therefore, quasi-sentences may contain only one word.

models for 100 epochs, where each epoch represents approximately 30,000 sentences.[12]

3.1 Evaluation

We evaluate 0-BlinC performance on predictions of policy areas in manifestos written in six out-of-sample target languages. Class predictions are made for each out-of-sample quasi-sentence and model fit is evaluated with accuracy and AUC, the area under the ROC. Model hyperparameters were originally tuned using the English-Spanish language pair; no changes were made to model hyperparameters before training on any additional language pairs.[13]

First, we turn to out-of-sample (test set) English classifier performance. For this test, we use the English-to-Spanish model. The class-wise confusion matrix (left) and ROC curve (right) are shown in Figure 2. The columns of a confusion matrix represent predicted class membership; the rows represent true class membership. The values are row-wise normalized so a single row represents the proportion of samples of a given class that are predicted to fall in each of the possible classes. Entries along the diagonal are correctly classified. The average classification accuracy for the English test set is 0.6, indicating that 60% of quasi-sentences are classified correctly.

We also include the ROC curve in Figure 2. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR). A ROC curve for a classifier that perfectly predicts class membership would form a 90 degree angle in the upper left of the plot – the TPR would be 1.0 while the FPR is 0.0. A classifier with no predictive power, one that performs randomly, would have a ROC curve that follows the diagonal $x = y$ (indicated by the dashed line). In interpreting ROC curves, it is instructive to consider the area under the curve (AUC). An AUC of 1.0 corresponds to perfect classification and an AUC of 0.5 is random classification. We plot the class-wise ROC curves and include their corresponding AUC values in the plot legend. For the out-of-sample English quasi-sentences, the model produces AUC values of between 0.79 and 0.93, depending on class. Also listed are the micro- and macro-average AUC scores corresponding to sample-wise and class-wise average AUC values, respectively; both are 0.86.

We now turn to out-of-sample (test set) performance in the six target languages for which we had

[12] We are abusing the term epoch which typically refers to a single pass over the entire dataset. Setting a maximum epoch size allows us to better manage memory and input/output constraints.

[13] With the exception of training time; due to an oversight, English-Spanish was trained for 150 epochs while the others were trained for only 100. In our experience, 0-BlinC typically converges well before even 100 epochs and so we expect the English-Spanish performance metrics are still comparable to those of the other language pairs.

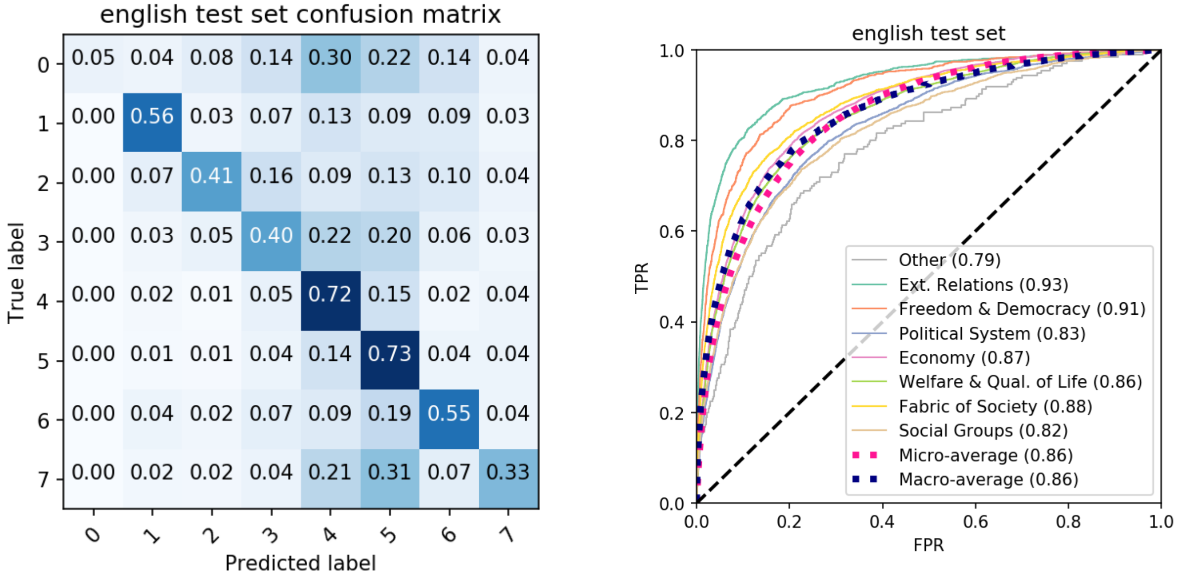


Figure 2: 0-BlinC Out-of-Sample Performance in English

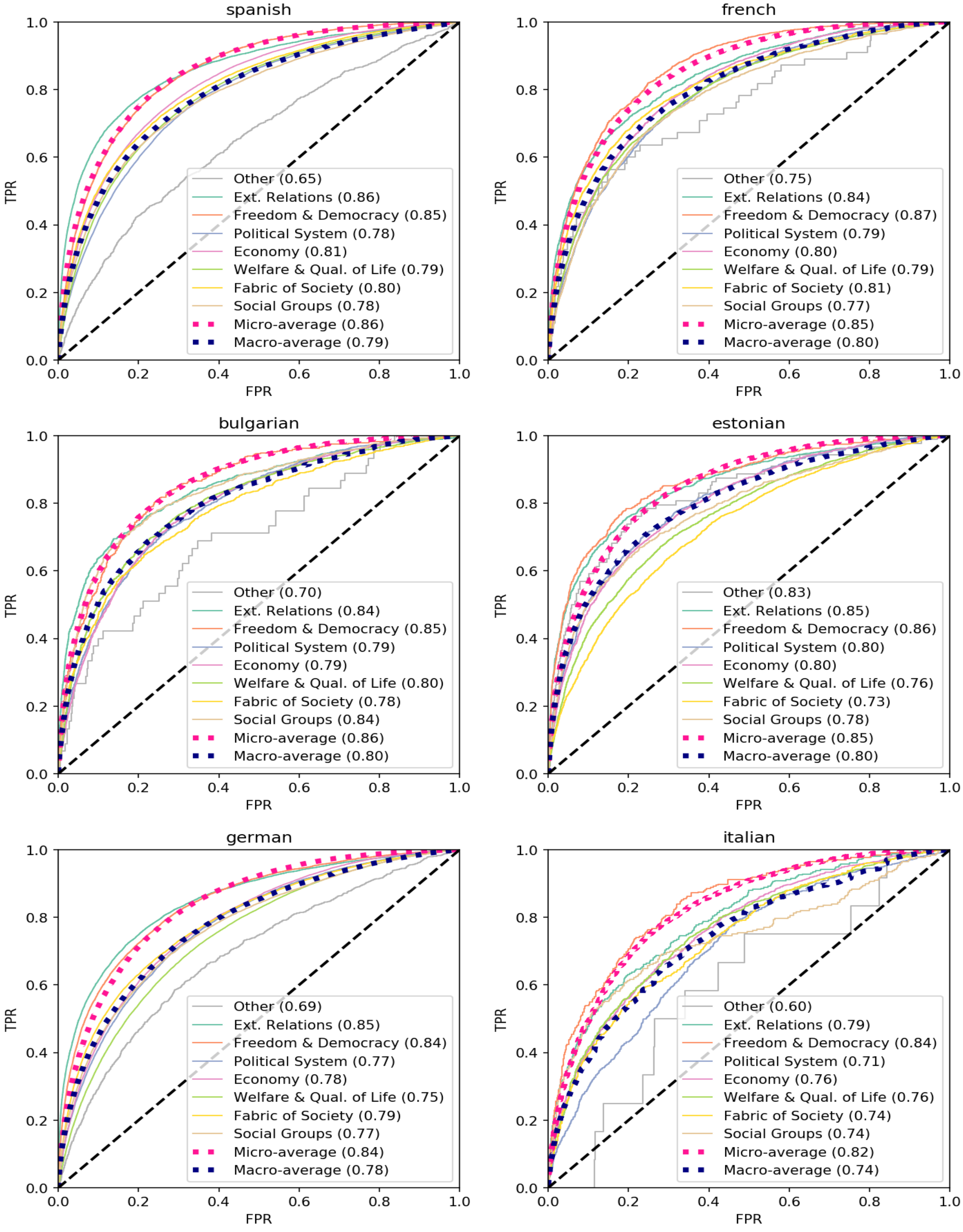
no labelled training data. Figure 3 shows the class-wise ROC curves and their corresponding AUC values for each of the six languages. Target language confusion matrices are also provided in Figure 4. Accuracy for non-English languages is between 0.43 and 0.5. There is a performance gap in accuracy of 10 to 17 percentage points from the English test set to the foreign language sets. This gap is the result of not only the quality of our language alignment model but also differences in manifesto content, salient issues across countries, and variation due to the original manifesto coders. Micro- and macro-average AUC scores fall between 0.82-0.86 and 0.74-0.80, respectively. Micro-average AUC scores tend to be higher than their corresponding macro-average AUC scores, indicative of better predictive performance on larger classes.[14] This is reflected in the relatively higher accuracy and AUC values associated with classes 4 and 5, *economy* and *welfare & quality of life*, two of the largest classes with respect to quasi-sentence volume. Class 0, *other*, is relatively uncommon and 0-BlinC performs poorly here in all languages.

3.2 Alternative Methods

0-BlinC is also compared against alternative approaches for text classification in languages for which no labelled data exist. Specifically, we compare 0-BlinC against Machine Translation (MT) and Pre-trained

[14] Micro-average statistics treat every data point equally in computation. Macro-average statistics compute the statistic of interest by class and then take the un-weighted average value of that statistic across classes.

Figure 3: 0-BlinC Out-of-Sample Performance in Six Languages



word embedding (PT). In the MT test, up to 20,000 quasi-sentences per language are translated into English using Google Translate.[15] A single English word2vec model and text classifier is then trained and applied to every translated sentence. All model parameters for the MT model are identical to those of 0-BlinC except that the MT model lacks an alignment component and a word2vec model in *language B*. The MT model is trained on the same English source data as 0-BlinC and all machine-translated sentences are held out until the evaluation stage (i.e. "out-of-sample").

In a second test, PT, pre-trained bilingual aligned word vectors are acquired from Facebook Research's MUSE project (Facebook Research, 2018). Rather than training four sub-tasks, we simply "translate" all words in all texts into their corresponding MUSE word vectors. Because the word vector and vector alignment tasks have already been accomplished by MUSE, we train only a text classifier that inputs word vectors and outputs class predictions. This classifier is identical to the classification component of 0-BlinC (sans the word2vec *A*, word2vec *B*, and alignment sub-models).

Table 1: Out-of-Sample Performance Comparison

	Accuracy			Micro-avg AUC			Macro-avg AUC		
	0-BlinC	PT	MT	**0-BlinC**	PT	MT	**0-BlinC**	PT	MT
English Test	0.60	–	–	0.86	–	–	0.86	–	–
Bulgarian	**0.50**	0.48	0.46	**0.86**	0.84	0.72	**0.80**	0.78	0.74
Estonian	0.48	0.43	**0.51**	**0.85**	0.81	0.74	**0.80**	0.74	**0.80**
French	**0.48**	0.44	0.42	**0.85**	0.82	0.72	**0.80**	0.75	0.73
German	**0.46**	0.42	0.43	**0.84**	0.81	0.70	**0.78**	0.75	0.73
Italian	**0.43**	0.37	0.34	**0.82**	0.78	0.72	**0.74**	0.71	0.67
Spanish	**0.49**	0.45	0.40	**0.86**	0.83	0.72	**0.79**	0.76	0.73

The performance metrics of MT and PT with respect to the manifesto classification task are presented alongside the results of 0-BlinC in Table 1. 0-BlinC outperforms both alternative methods in nearly all cases. The only exceptions are in Estonian where the Google Translate-based MT model ties with or slightly outperforms 0-BlinC in macro-average AUC and multiclass accuracy, respectively. For all other languages and all other evaluation metrics, 0-BlinC outperforms both MT and PT.

[15] We actually translate $min(20000, n)$ where n is the number of sentences in the source language. A small number of sentences are lost because Google Translate failed to return a result. Bulgarian: 10,654 quasi-sentences, German: 19,976, Spanish: 19,984, Estonian: 11,163, French: 19,995, and Italian: 4,175.

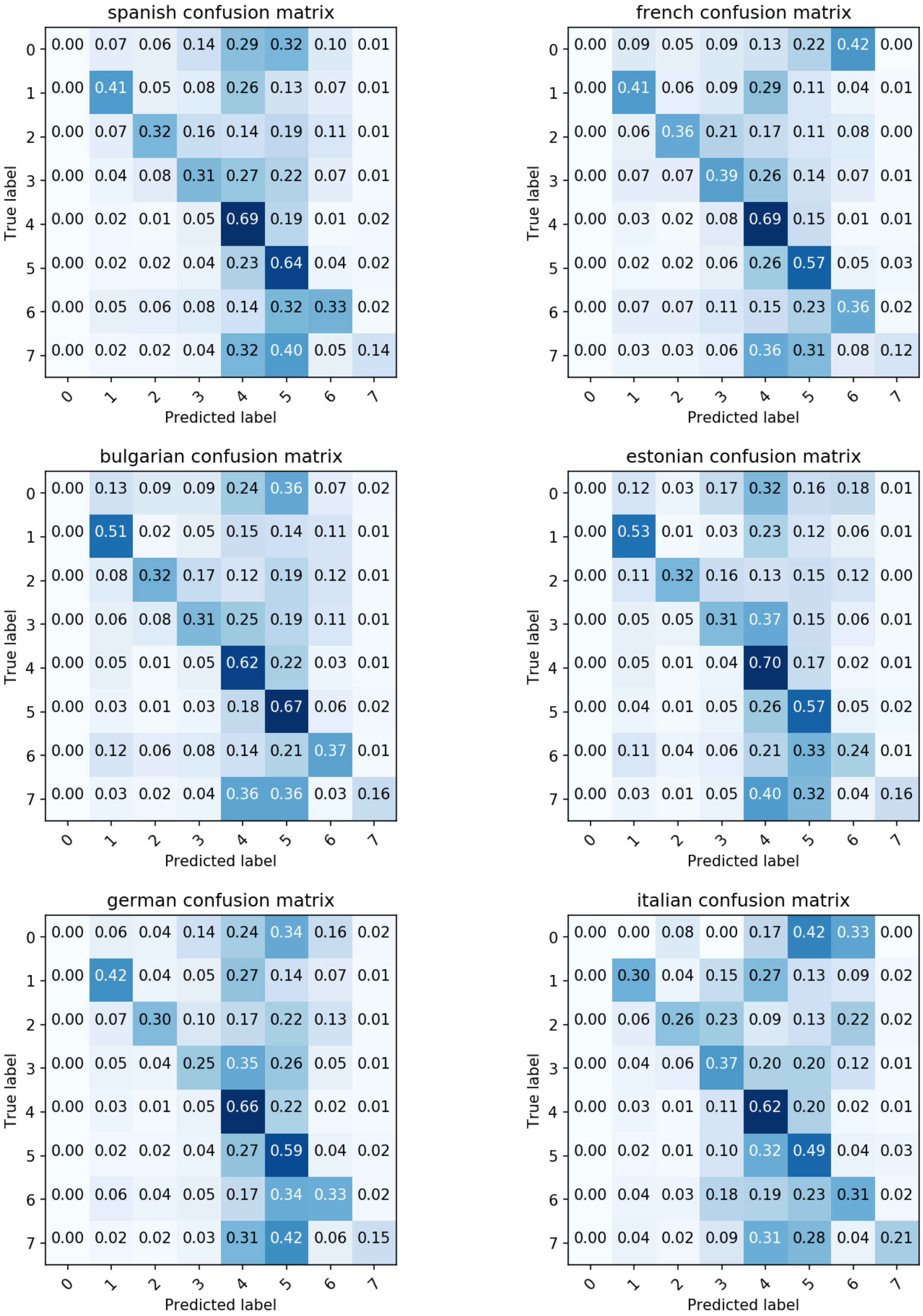
4 Conclusion

We introduce a novel multi-task neural network that produces bilingual word embeddings, performs document classification in a source language, and is able to classify documents in a target language without any classification training data in that language. By optimizing the classification, word2vec, and alignment models jointly, the 0-BlinC is able to transfer knowledge learned by classifying documents in one language to enable classification of documents in a different language. Using the supervised learning task of classifying policy issue areas of party manifestos in 7 languages, we show that our method is able to produce aligned word vectors for a bilingual corpus and is able to predict policy areas in out-of-sample documents with AUC values of around 0.85. The performance of 0-BlinC is stable and evaluation metrics are similar across several languages from multiple language families.

Although 0-BlinC seems complicated, each sub-model is simple shallow neural network. There are also several extensions that might improve the model's performance. First, we achieved impressive model performance using minimal text pre-processing. It is possible that improved language-specific pre-processing of the texts could lead to improved model performance. Second, we hope to compare our word alignment method to existing methods, such as BilBOWA (Gouws, Bengio and Corrado, 2015). Third, 0-BlinC does not take into account the order of words within sentences or documents. In the next step, we can use recurrent and convolution-based variants to incorporate syntactic information. Finally, we hope to evaluate whether a 0-BlinC variant that includes training labels from multiple languages might perform better than any single model trained on a single language.

The structure of our proposed model is highly flexible. It is possible to extend the current model to more than two languages by adding additional word2vec and alignment sub-models. Because the embedding learned by the word2vec models is a function of not only the standard word2vec loss term but also the classifier loss term, we anecdotally observe that the word embeddings tend to reflect both language properties and characteristics of the classes in the classifier. We expect that additional exploration of this could facilitate easier-to-interpret word vectors or semi-supervised word-vectors, similar to structural topic modeling (Roberts et al., 2014). We believe the our proposed approach can benefit a broad set of quantitative text analysis research. We will make our 0-BlinC model available as a Python package soon to enable easy application of 0-BlinC.

Figure 4: 0-BlinC Target Language Confusion Matrices



spanish confusion matrix

	0	1	2	3	4	5	6	7
0	0.00	0.07	0.06	0.14	0.29	0.32	0.10	0.01
1	0.00	0.41	0.05	0.08	0.26	0.13	0.07	0.01
2	0.00	0.07	0.32	0.16	0.14	0.19	0.11	0.01
3	0.00	0.04	0.08	0.31	0.27	0.22	0.07	0.01
4	0.00	0.02	0.01	0.05	0.69	0.19	0.01	0.02
5	0.00	0.02	0.02	0.04	0.23	0.64	0.04	0.02
6	0.00	0.05	0.06	0.08	0.14	0.32	0.33	0.02
7	0.00	0.02	0.02	0.04	0.32	0.40	0.05	0.14

french confusion matrix

	0	1	2	3	4	5	6	7
0	0.00	0.09	0.05	0.09	0.13	0.22	0.42	0.00
1	0.00	0.41	0.06	0.09	0.29	0.11	0.04	0.01
2	0.00	0.06	0.36	0.21	0.17	0.11	0.08	0.00
3	0.00	0.07	0.07	0.39	0.26	0.14	0.07	0.01
4	0.00	0.03	0.02	0.08	0.69	0.15	0.01	0.01
5	0.00	0.02	0.02	0.06	0.26	0.57	0.05	0.03
6	0.00	0.07	0.07	0.11	0.15	0.23	0.36	0.02
7	0.00	0.03	0.03	0.06	0.36	0.31	0.08	0.12

bulgarian confusion matrix

	0	1	2	3	4	5	6	7
0	0.00	0.13	0.09	0.09	0.24	0.36	0.07	0.02
1	0.00	0.51	0.02	0.05	0.15	0.14	0.11	0.01
2	0.00	0.08	0.32	0.17	0.12	0.19	0.12	0.01
3	0.00	0.06	0.08	0.31	0.25	0.19	0.11	0.01
4	0.00	0.05	0.01	0.05	0.62	0.22	0.03	0.01
5	0.00	0.03	0.01	0.03	0.18	0.67	0.06	0.02
6	0.00	0.12	0.06	0.08	0.14	0.21	0.37	0.01
7	0.00	0.03	0.02	0.04	0.36	0.36	0.03	0.16

estonian confusion matrix

	0	1	2	3	4	5	6	7
0	0.00	0.12	0.03	0.17	0.32	0.16	0.18	0.01
1	0.00	0.53	0.01	0.03	0.23	0.12	0.06	0.01
2	0.00	0.11	0.32	0.16	0.13	0.15	0.12	0.00
3	0.00	0.05	0.05	0.31	0.37	0.15	0.06	0.01
4	0.00	0.05	0.01	0.04	0.70	0.17	0.02	0.01
5	0.00	0.04	0.01	0.05	0.26	0.57	0.05	0.02
6	0.00	0.11	0.04	0.06	0.21	0.33	0.24	0.01
7	0.00	0.03	0.01	0.05	0.40	0.32	0.04	0.16

german confusion matrix

	0	1	2	3	4	5	6	7
0	0.00	0.06	0.04	0.14	0.24	0.34	0.16	0.02
1	0.00	0.42	0.04	0.05	0.27	0.14	0.07	0.01
2	0.00	0.07	0.30	0.10	0.17	0.22	0.13	0.01
3	0.00	0.05	0.04	0.25	0.35	0.26	0.05	0.01
4	0.00	0.03	0.01	0.05	0.66	0.22	0.02	0.01
5	0.00	0.02	0.02	0.04	0.27	0.59	0.04	0.02
6	0.00	0.06	0.04	0.05	0.17	0.34	0.33	0.02
7	0.00	0.02	0.02	0.03	0.31	0.42	0.06	0.15

italian confusion matrix

	0	1	2	3	4	5	6	7
0	0.00	0.00	0.08	0.00	0.17	0.42	0.33	0.00
1	0.00	0.30	0.04	0.15	0.27	0.13	0.09	0.02
2	0.00	0.06	0.26	0.23	0.09	0.13	0.22	0.02
3	0.00	0.04	0.06	0.37	0.20	0.20	0.12	0.01
4	0.00	0.03	0.01	0.11	0.62	0.20	0.02	0.01
5	0.00	0.02	0.01	0.10	0.32	0.49	0.04	0.03
6	0.00	0.04	0.03	0.18	0.19	0.23	0.31	0.02
7	0.00	0.04	0.02	0.09	0.31	0.28	0.04	0.21

References

Benoit, Kenneth, Kevin Munger and Arthur Spirling. 2017. "Measuring and Explaining Political Sophistication Through Textual Complexity.".

Boschee, Elizabeth, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz and Michael Ward. 2015. "ICEWS coded event data." *Harvard Dataverse* 12.

Chollet, François et al. 2015. "Keras." https://keras.io.

Crabtree, Charles, Matt Golder, Thomas Gschwend and Indriði Indriðason. 2018. "It's Not Only What you Say, It's Also How You Say It: The Strategic Use of Campaign Sentiment." Working Paper.

Facebook Research. 2018. "MUSE: Multilingual Unsupervised and Supervised Embeddings.".
URL: *https://github.com/facebookresearch/MUSE*

Gouws, Stephan, Yoshua Bengio and Greg Corrado. 2015. "BilBOWA: Fast Bilingual Distributed Representations without Word Alignments." *Proceedings of the 32nd International Conference on Machine Learning* .

Hawkins, Kirk A. and Bruno Castanho Silva. 2018. Textual Analysis: Big Data Approaches. In *The Ideational Approach to Populism: Concept, Theory, and Analysis*, ed. Kirk A. Hawkins, Ryan E. Carlin, Levente Littvay and Cristóbal Rovira Kaltwasser. New York: Routledge pp. 44–74.

King, Gary, Jennifer Pan and Margaret E Roberts. 2017. "How the Chinese government fabricates social media posts for strategic distraction, not engaged argument." *American Political Science Review* 111(3):484–501.

Kingma, Diederik P. and Jimmy Lei Ba. 2015. "Adam: A Method for Stochastic Optimization." *ICLR* .

Koehn, Philipp. 2005. "Europarl: A Parallel Corpus for Statistical Machine Translation." *MT Summit* .

Lowe, Will, Kenneth Benoit, Slava Mikhaylov and Michael Laver. 2011. "Scaling policy preferences from coded political texts." *Legislative studies quarterly* 36(1):123–155.

Lucas, Christopher, Richard A Nielsen, Margaret E Roberts, Brandon M Stewart, Alex Storer and Dustin Tingley. 2015. "Computer-assisted text analysis for comparative politics." *Political Analysis* 23(2):254–277.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and their Compositionality." *arXiv:1310.4546* .

Mikolov, Tomas, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *arXiv:1301.3781* .

Pan, Jennifer and Kaiping Chen. 2018. "Concealing Corruption: How Chinese Officials Distort Upward Reporting of Online Grievances." *American Political Science Review* pp. 1–19.

Peterson, Andrew and Arthur Spirling. 2018. "Classification Accuracy as a Substantive Quantity of Interest: Measuring Polarization in Westminster Systems." *Political Analysis* 26(1):120–128.

Powers, David M. 1998. "Applications and Explanations of Zipf's Law." *New Methods in Language Processing and Computational Natural Language Processing* pp. 151–160.

Roberts, Margaret E, Brandon M Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson and David G Rand. 2014. "Structural topic models for open-ended survey responses." *American Journal of Political Science* 58(4):1064–1082.

Rooduijn, Matthijs, Sarah L. de Lange and Wouter van der Brug. 2014. "A populist zeitgeist? Programmatic contagion by populist parties in Western Europe." *Party Politics* 20(4):563–575.

Ruder, Sebastian, Ivan Vulić and Anders Søgaard. 2017. "A Survey of Cross-lingual Word Embedding Models." *arXiv:1706.04902* .

Schrodt, Philip A. 2000. "Pattern Recognition of International Crises Using." *Political complexity: Nonlinear models of politics* p. 296.

Slapin, Jonathan B and Sven-Oliver Proksch. 2008. "A scaling model for estimating time-series party positions from texts." *American Journal of Political Science* 52(3):705–722.

Volkens, Andrea, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel and Bernhard Weßels. 2018. "The Manifesto Data Collection." *Manifesto Project (MRG/CMP/MARPOR)* .