Cross-Lingual Classification of Topics in Political Texts

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto
Data and Web Science Group
Faculty of Business Informatics and Mathematics
University of Mannheim
B6, 26, DE-68159, Mannheim, Germany
{goran, federico, simone}@informatik.uni-mannheim.de

Abstract

In this paper, we propose an approach for cross-lingual topical coding of sentences from electoral manifestos of political parties in different languages. To this end, we exploit continuous semantic text representations and induce a joint multilingual semantic vector spaces to enable supervised learning using manually-coded sentences across different languages. Our experimental results show that classifiers trained on multilingual data yield performance boosts over monolingual topic classification.

1 Introduction

Political parties are at the core of contemporary democratic systems. Election programs (the so-called *manifestos*), in which parties declare their positions over a range of topics (e.g., foreign policies, welfare, economy), are a widely used information source in political science. Within the Comparative Manifesto Project (CMP) (Volkens et al., 2011), political scientists have been collecting and topically coding manifestos from countries around the world for almost two decades now.

Manual topic coding of manifesto sentences, following the Manifesto Coding scheme with more than fifty fine-grained topics, grouped in seven coarse-grained topics (e.g, External Relations, Economy),[1] is time consuming and requires expert knowledge (King et al., 2017). Moreover, it is difficult to ensure annotation consistency, especially across different countries and languages (Mikhaylov et al., 2012). Nonetheless, manually coded manifestos remain the crucial data source for studies in computational political science (Lowe et al., 2011; Nanni et al., 2016).

[1] https://manifestoproject.wzb.eu/coding_schemes/mp_v5

In order support manual coders and mitigate the issues pertaining to manual coding, researchers have employed automatic text classification to topically label political texts (Karan et al., 2016; Zirn et al., 2016). Existing classification models utilize discrete representation of text (i.e., bag of words) and can thus exploit only monolingual data (i.e., train and predict same language instances).

In contrast, in this work, we aim to exploit multilingual data – topically-coded CMP manifestos in different languages. We propose a classification model that can be trained on multilingual corpus of political texts.To this effect, we induce semantic representations of texts from ubiquitous word embeddings (Mikolov et al., 2013b; Pennington et al., 2014) and induce a joint multilingual embedding space via the linear translation matrices (Mikolov et al., 2013a). We then experiment with two classification models, support vector machines (SVM) and convolutional neural network (CNN) that use embeddings from the joint multilingual space as input. Experimental results offer evidence that topic classifiers leveraging multilingual training sets outperform monolingual classifiers.

2 Related Work

The recent adoption of NLP methods had led to significant advances in the field of Computational Social Science (CSS) (Lazer et al., 2009) and political science in particular (Grimmer and Stewart, 2013). Among other tasks, researchers have addressed the identification of political differences from text (Sim et al., 2013; Menini and Tonelli, 2016), positioning of political entities on a left-right spectrum (Slapin and Proksch, 2008; Glavaš et al., 2017), as well as the detection of political events (Nanni et al., 2017) and prominent topics (Lauscher et al., 2016) in political texts.

For what concerns the analysis of manifestos,

previous studies have focused on topical segmentation (Glavaš et al., 2016) and monolingual (English) classification of sentences into coarse-grained topics (Zirn et al., 2016). Because manifesto sentences are short and short text classification is inherently challenging due to limited context, Zirn et al. (2016) proposed to apply a global optimization step (performed via Markov Logic network) on top of independent topic decisions for sentences. Numerous supervised models have also been proposed for classification of other types of political text (Purpura and Hillard, 2006; Stewart and Zhukov, 2009; Verberne et al., 2014; Karan et al., 2016, *inter alia*). However, these models also represent texts as sets of discrete words which directly limits their applicability to monolingual classification settings only.

3 Cross-lingual Classification

We first explain how we induce the joint multilingual embedding space and then describe the two classification models we experimentally evaluated.

3.1 Multilingual Embedding Space

Words from different languages can be semantically compared only if their embeddings come from the same multidimensional semantic space. However, independent training of monolingual word embeddings, as obtained by running embedding models (Mikolov et al., 2013b; Pennington et al., 2014) on large monolingual corpora, will result in completely unassociated spaces between the languages (e.g., the English embedding of *"bad"* will not be similar to the German embedding of *"schlecht"*).

Consequently, to enable a unified representation of texts in different languages, we must first map different monolingual embedding spaces to a joint multilingual space in which words from different languages will become semantically comparable. To this end, we set the semantic space of one language as the *target embedding space*) and translate vectors of all words from all other languages to the target space. The translation is performed using the linear translation model proposed by Mikolov et al. (2013a), who observed that there exists a linear translation between embedding spaces independently trained on different corpora.

Given a set of N word translations pairs $\{w_{s_i}, w_{t_i}\}_{i=1}^{N}$, we learn a translation matrix $\mathbf{M}$ that projects the embedding vectors from the *source space* to the *target space*. Let $\mathbf{S}$ be the matrix composed of embeddings of all source words w_{s_i} from translation pairs and $\mathbf{T}$ be the matrix made of embeddings of corresponding target words w_{t_i}. Unlike the original work (Mikolov et al., 2013a), and following the observations from Glavaš et al. (2017), we do not learn the translation matrix $\mathbf{M}$ via iterative numeric optimization, but analytically by multiplying the Moore-Penrose pseudoinverse of the source matrix $\mathbf{S}$ ($\mathbf{S}^{+}$) with the target matrix $\mathbf{T}$, i.e., $\mathbf{M} = \mathbf{S}^{+} \cdot \mathbf{T}$. The translation matrices obtained via the pseudoinverse seem to be of same quality as those obtained through numeric optimization (Glavaš et al., 2017).

3.2 Classification Models

We experiment with two classification models that are able to take text embeddings as input for classification – SVM and CNN. Taking embeddings as input, models are fully agnostic of the language of text instances. Therefore, we must ensure that representations of all instances are translated to the joint multilingual embedding space before we feed them to the classifiers.

3.2.1 Convolutional Neural Network

Recently, convolutional neural networks (LeCun and Bengio, 1998, CNN) have yielded best performance on many text classification tasks (Kim, 2014; Severyn and Moschitti, 2015). CNN is a feed-forward neural network consisting of one or more convolution layers. Each convolution layer consists of a set of filters matrices (parameters of the model optimized during training). In text classification, the convolution operation is computed sequentially between each filter matrix and each slice (of the same size as filter) of the embedding matrix representing the input text. Each convolution layer is coupled with a pooling layer, in which only the subset of largest convolution scores produced by each filter is retained and used as input either for the next convolution layer or the final fully-connected prediction layer. With such architecture, CNN captures local aspects of texts, i.e., the most informative k-grams (where k is the filter size) in the input text with respect to the classification task. Following previous work (Kim, 2014; Severyn and Moschitti, 2015), we train CNNs with a single convolution and single pooling layer.

The input representation of each text instance for the CNN is a sequence of word embeddings – i.e., each text instance is represented with a $N \times K$ matrix, with N being the length of the text and K the length of word embeddings. CNN requires

the input matrices to have the same size for all training instances. Thus, all text instances must be adjusted so that they are of the same length. In all our experiments, we set N to the number of tokens of the longest text in the dataset. We then pad all other sentences with a special padding token (which is assigned a random embedding vector), in order to make them N tokens long as well.

3.2.2 SVM with Sentence Embeddings

The second model we employ is SVM classifier. Since (1) SVMs, unlike CNN, cannot take a matrix as input and (2) concatenating embedding vectors of sentence words into one large embedding vector would result in a too large feature space, we first compute the aggregated embedding vector of the sentence from the embeddings of its constituent words and then feed this aggregate sentence embedding to the SVM classifier. The sentence embedding is a weighted continuous bag of words (WCBOW) aggregation of word embeddings:

$$WCBOW(t_1, \ldots, t_k) = \frac{1}{\sum_{i=1}^{k} w_i} \sum_{i=1}^{k} w_i e(t_i)$$

where t_i is the i-th token of the input text, $e(t_i)$ is the word embedding of the token t_i, and weight w_i is the TF-IDF score of the token-sentence pair, used to assign more importance to more informative words. Considering that the resulting sentence embedding is a low-dimensional (e.g., 100 dimensions) dense numeric vector, we opted for the SVM classifier with non-linear RBF kernel.

4 Evaluation

We first describe the multilingual dataset of manually topically-coded manifestos. We then describe the experimental setting and finally present and discuss the results.

4.1 Dataset

We collected all available manually topically-coded manifestos in four different languages: English (20196 annotated sentences), French (4808), German (48117), and Italian (4370). In order to compare the results across languages more clearly, we opted for a language-balanced dataset, containing the same number of instances in all four languages. Thus, we randomly sampled 4370 (number of annotated sentences in Italian, the lowest number across the four languages) sentences from English, French, and German manifestos. The distribution of sentences over the seven coarse-grained manifesto topics in the obtained dataset is shown in Table 1. We next split the dataset into the train, development, and test portion (70%-15%-15% ratio).[2]

Topic	% of Sentences
External Relations	10%
Freedom & Democracy.	8%
Political System	10%
Economy	24%
Welfare & Quality of Life	28%
Fabric of Society	11%
Social Groups	9%

Table 1: Topic distribution in the dataset.

Translation	P@1 (%)	P@5 (%)
DE $\rightarrow$ EN	31.6	52.6
FR $\rightarrow$ EN	38.3	55.6
IT $\rightarrow$ EN	34.4	50.8

Table 2: Quality of translation matrices.

4.2 Experimental Setting

Embeddings and translation matrices. We obtained the pre-trained monolingual word embeddings for all four languages: CBOW embeddings (Mikolov et al., 2013b) for German (100 dim.), Italian (300 dim.), and French (300 dim.) and GloVe embeddings (Pennington et al., 2014) for English (100 dim.). We created the multilingual embedding space by mapping embeddings of other three languages to the English embedding space.[3]

We obtained the word translation pairs, required to learn the translation matrices by translating 4200 most frequent English words to the other three languages using Google Translate. We then used 4000 pairs to train each of the translation matrices (DE $\rightarrow$ EN, FR $\rightarrow$ EN, and IT $\rightarrow$ EN) and remaining 200 pairs for evaluation of translation quality. The quality of obtained translation matrices is shown in Table 2 in terms of P@1 and P@5.

Evaluation settings. Our primary goal is to evaluate whether the cross-lingual models, which are able to use instances in different languages for training perform better than models using only instances

[2]We make the dataset freely available at https://tinyurl.com/ml835s8

[3]Glavaš et al. (2017) showed that using monolingual embeddings of different sizes trained with different algorithms has no negative effect on learned translation matrices.

Setting	Model	EN	DE	FR	IT
Mono-L	Linear SVM (BoW)	.54	**.44**	.63	.53
	SVM RBF (emb)	.43	.31	.42	.37
	CNN	.57	.41	.59	.33
Cross-L	SVM RBF (emb)	.30	.30	.49	.40
	CNN	**.59**	.40	**.86**	**.84**

Table 3: Topic classification results.

from one language (i.e., train and test sentences of same language). To this end, we evaluate both models, SVM and CNN, in both the *monolingual* and *cross-lingual* setting. In the monolingual setting (Mono-L), the models are respectively trained, optimized, and evaluated on train, validation, and test instances of the same language. In the cross-lingual setting (Cross-L), we train the models on the union of training instances of all four languages. On one hand, the Cross-L training set is four times larger than each individual Mono-L training set. On the other hand, instances of the same topic should be more heterogeneous as they (1) originate from different languages and (2) were obtained via imperfect embedding translation (except for English). In addition to the models from Section 3.2, in the Mono-L setting, as a baseline, we evaluate a simple linear SVM with bag-of-words features.

Model optimization. We learn the CNN parameters using the RMSProp algorithm (Tieleman and Hinton, 2012). In all experiments, we optimize the models' hyperparameters (C and γ for RBF kernel SVM, filter sizes, number of filters, and dropout rate for CNN) on the corresponding (monolingual) validation portion of the dataset. We then report the performance of the model with optimal hyperparameter values on the corresponding (monolingual) test set.

4.3 Results and Discussion

In Table 3 we show the topic classification performance of the models, in terms of F_1 score (micro-averaged over all seven topic classes). Considering the predictions for individual topics, all models, unsurprisingly, yielded best performance for the two classes with largest number of instances in training sets: *Economy* and *Welfare & Quality of Life*.

In the monolingual setting (Mono-L), surprisingly, the baseline SVM using lexical features seems to perform better than both embedding-based RBF-kernel SVM and CNN. Since the RBF-kernel SVM with aggregate embedding features displays poor performance in the cross-lingual setting as well, we speculate that the aggregate sentence embeddings are semantically too fuzzy (especially for long sentences) and consequently less informative for discriminating the political topics. On the other hand, CNN shows improvements in performance when trained using the multilingual training set (for all languages except German). We believe that the monolingual training sets are simply too small to successfully learn the good values for CNN parameters. Cross-L performance of CNN models shows the benefits of using multilingual training data for topic classification, enabled through the induction of the joint multilingual embedding space.

We observe that the Cross-L prediction performance across languages varies dramatically. When trained on Cross-L training set, CNN shows small prediction improvement for English, no improvement for German, and drastic improvements for French and Italian. We believe that this large variance across languages can be credited to different levels of (in)consistency in manual topic annotations. Political scientists working with CMP data have already observed substantial inconsistencies in manual topic coding of manifestos (Mikhaylov et al., 2012; Gemenis, 2013). Our results suggest that German and English annotations are significantly less consistent than French and Italian. CMP started coding French and Italian manifestos only recently (in 2012 and 2013, respectively), whereas the German and English manifestos have been coded for almost two decades. Being coded over a much longer period of time, German and English manifestos (1) cover a wider span of political issues (with more language variation) and (2) have been coded by a larger number of coders over the years. Both these factors inevitably lead to less consistent topic annotations. Additional inconsistency for English manifestos possibly stems from different countries of their origin (USA, UK).

5 Conclusion

In this paper we proposed an approach for automated cross-lingual topical coding of political manifestos. We exploit continuous semantic text representations (i.e., embeddings) and induce a joint multilingual spaces, allowing us to train topic classifiers on manually coded data from different languages. Obtained experimental results show that the classifiers trained on a multilingual data outperform monolingual topic classifiers.

References

Kostas Gemenis. 2013. What to do (and not to do) with the comparative manifestos project data. *Political Studies* 61(1 suppl):3–23.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised text segmentation using semantic relatedness graphs. In **SEM*. pages 125–130.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. Unsupervised cross-lingual scaling of political texts. In *EACL*.

Justin Grimmer and Brandon M Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis* 21(3):267–297.

Mladen Karan, Daniela Širinić, Jan Šnajder, and Goran Glavaš. 2016. Analysis of policy agendas: Lessons learned from automatic topic classification of croatian political texts. In *LaTeCh*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. Doha, Qatar.

Gary King, Patrick Lam, and Margaret Roberts. 2017. Computer assisted keyword and document set discovery from unstructured text. *American Journal of Political Science* .

Anne Lauscher, Federico Nanni, Pablo Ruiz Fabo, and Simone Paolo Ponzetto. 2016. Entities as topic labels: combining entity linking and labeled lda to improve topic interpretability and evaluability. *Italian Journal of Computational Linguistics* 2(2):67–88.

David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. 2009. Life in the network: the coming age of computational social science. *Science (New York, NY)* 323(5915):721.

Yann LeCun and Yoshua Bengio. 1998. The handbook of brain theory and neural networks. MIT Press, Cambridge, MA, USA, chapter Convolutional Networks for Images, Speech, and Time Series.

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

Stefano Menini and Sara Tonelli. 2016. Agreement and disagreement: Comparison of points of view in the political domain. In *Coling*. pages 2461–2470.

Slava Mikhaylov, Michael Laver, and Kenneth R Benoit. 2012. Coder reliability and misclassification in the human coding of party manifestos. *Political Analysis* .

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR* .

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

Federico Nanni, Simone Paolo Ponzetto, and Laura Dietz. 2017. Building entity-centric event collections. JCDL.

Federico Nanni, Cäcilia Zirn, Goran Glavaš, Jason Eichorst, and Simone Paolo Ponzetto. 2016. Topfish: topic-based analysis of political position in us electoral campaigns. In *PolText*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. pages 1532–1543.

Stephen Purpura and Dustin Hillard. 2006. Automated classification of congressional legislation. In *Proceedings of the 2006 international conference on Digital government research*. Digital Government Society of North America, pages 219–225.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *SIGIR*.

Yanchuan Sim, Brice Acree, Justin H Gross, and Noah A. Smith. 2013. Measuring ideological proportions in political speeches. In *EMNLP*.

Jonathan B Slapin 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.

Brandon M Stewart and Yuri M Zhukov. 2009. Use of force and civil–military relations in russia: an automated content analysis. *Small Wars & Insurgencies* 20(2):319–343.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. Technical Report 2.

Suzan Verberne, Eva Dhondt, Antal van den Bosch, and Maarten Marx. 2014. Automatic thematic classification of election manifestos. *Information Processing & Management* 50(4):554–567.

Andrea Volkens, Onawa Lacewell, Pola Lehmann, Sven Regel, Henrike Schultze, and Annika Werner. 2011. The manifesto data collection. *Manifesto Project (MRG/CMP/MARPOR), Berlin: Wissenschaftszentrum Berlin für Sozialforschung (WZB)* .

Cäcilia Zirn, Goran Glavaš, Federico Nanni, Jason Eichorts, and Heiner Stuckenschmidt. 2016. Classifying topics and detecting topic shifts in political manifestos. In *PolText*.