Joint Sentence–Document Model for Manifesto Text Analysis

Shivashankar Subramanian **Trevor Cohn** **Timothy Baldwin** **Julian Brooke**
School of Computing and Information Systems
University of Melbourne
shivashankar@student.unimelb.edu.au, {t.cohn, tbaldwin, julian.brooke}@unimelb.edu.au

Abstract

Election manifestos document the intentions, motives, and views of political parties. They are often used for analysing party policies and positions on various issues, as well as for quantifying a party's position on the left–right spectrum. In this paper we propose a model for automatically predicting both types of analysis from manifestos, based on a joint sentence–document approach which performs both sentence-level thematic classification and document-level position quantification. Our method handles text in multiple languages, via the use of multilingual vector-space embeddings. We empirically show that the proposed joint model performs better than state-of-art approaches for the document-level task and provides comparable performance for the sentence level task, using manifestos from thirteen countries, written in six different languages.

1 Introduction

Election manifestos are a core artifact in political text analysis. One of the widely used datasets by political scientists is the Comparative Manifesto Project (CMP) dataset, initiated by Volkens et al. (2011), that collects party manifestos from elections in many countries around the world. The goal of the project is to provide a large data collection to support political studies on electoral processes. A sub-part of the manifestos has been manually annotated at the sentence-level with one of over fifty fine-grained political themes, divided into 7 coarse-grained topics (see Table 5). These are important because it can be seen as party positions on fine-grained policy themes and also the coded text can be used for various downstream tasks (Lowe et al., 2011). While manual annotations are very useful for political analyses, they come with two major drawbacks. First, it is very time-consuming and labor-intensive to manually annotate each sentence with the correct category from a complex annotation scheme. Secondly, coder preferences towards particular categories might lead to annotation inconsistencies and affect comparability between manifestos annotated by different coders (Mikhaylov et al., 2012). In order to overcome these challenges, fine and coarse-level manifesto sentence classification was addressed using supervised machine learning techniques (Verberne et al., 2014; Zirn et al., 2016). Nonetheless, manually-coded manifestos remain the crucial data source for studies in computational political science (Lowe et al., 2011; Nanni et al., 2016).

Other than the sentence-level labels, the manifesto text also has document-level signals, which quantify its position on the left–right spectrum (Slapin and Proksch, 2008). Though sentence-level classification and document-level quantification tasks are inter-dependent, existing work handles them separately. We instead propose a joint approach to model the two tasks together. Overall, the contributions of this work are as follows:

- we empirically study the utility of multilingual embeddings for cross-lingual manifesto text analysis — at the sentence (for 57-class classification) and document-levels (for RILE score regression)
- we evaluate the effectiveness of modelling the sentence- and document-level tasks together
- we study the value of *country* information used in conjunction with text for the

Shivashankar Subramanian, Trevor Cohn, Timothy Baldwin and Julian Brooke. 2017. Joint Sentence-Document Model for Manifesto Text Analysis. In *Proceedings of Australasian Language Technology Association Workshop*, pages 25–33.

document-level regression task.

2 Related Work

The recent adoption of NLP methods has led to significant advances in the field of Computational Social Science (Lazer et al., 2009), including political science (Grimmer and Stewart, 2013). Some popular tasks addressed with political text include: party position analysis (Biessmann, 2016); political leaning categorization (Akoglu, 2014; Zhou et al., 2011); stance classification (Sridhar et al., 2014); identifying keywords, themes & topics (Karan et al., 2016; Ding et al., 2011); emotion analysis (Rheault, 2016); and sentiment analysis (Bakliwal et al., 2013). The source data includes manifestos, political speeches, news articles, floor debates and social media posts.

With the increasing availability of large-scale datasets and computational resources, large-scale comparative political text analysis has gained the attention of political scientists (Lucas et al., 2015). For example, rather than analyzing the political manifestos of a particular party during an election, mining different manifestos across countries over time can provide deeper comparative insights into political change.

Existing classification models, except (Glavaš et al., 2017), utilize discrete representation of text (i.e., bag of words). Also, most of the work analyzes manifesto text at the country level. Recent work has demonstrated the utility of neural embeddings for multi-lingual coarse-level topic classification (7 major categories) over manifesto text (Glavaš et al., 2017). The authors show that multi-lingual embeddings are more effective in the cross-lingual setting, where labeled data is used from multiple languages. In this work, we focus on cross-lingual fine-grained thematic classification (57 categories in total), where we have labeled data for all the languages.

For the document-level quantification task, much work has used label count aggregation of manually-annotated sentences as features (Lowe et al., 2011; Benoit and Däubler, 2014), while other work has used dictionary- based supervised methods, or unsupervised factor analysis based techniques (Hjorth et al., 2015; Bruinsma and Gemenis, 2017). The latter method uses discrete word representations and deals with mono-lingual text only. In Glavas et al. (2017), the authors leverage neural embeddings for cross-lingual EU parliament speech text quantification with two pivot texts for extreme left and right positions. They represent the documents using word embeddings averaged with TF-IDF scores as weights. All these approaches model the sentence and document-level tasks separately.

3 Manifesto Text Analysis

In the CMP, trained annotators manually label manifesto sentences according to the 57 fine-grained political categories (shown in Table 5), which are grouped into seven policy areas: External Relations, Freedom and Democracy, Political System, Economy, Welfare and Quality of Life, Fabric of Society, and Social Groups. Political parties either write their promises as a bulleted list of individual sentences, or structured as paragraphs (an example is given in Figure 4), providing more information on topic coherence. Also the length of documents, measured as the number of sentences, varies greatly between manifestos. The typical length (in sentences) over manifestos (948 in total) from 13 countries — Austria, Australia, Denmark, Finland, France, Germany, Italy, Ireland, New Zealand, South Africa, Switzerland, United Kingdom and United States — is 516.7±667. Variance in the number of sentences across documents in conjunction with class imbalance makes automated thematic classification a challenging task.

While annotating, a sentence is split into multiple segments if it discusses unrelated topics or different aspects of a larger policy, e.g. (as indicated by the different colors, and associated integer labels):

> We need to address our close ties with our neighbours (107) as well as the unique challenges facing small business owners in this time of economic hardship. (402)

Such examples are not common, however.[1] Also the segmentation was shown to be inconsistent and to have no effect on quantifying the proportion of sentences discussing various topics and document-level regression tasks (Däubler et al., 2012). Hence, consistent with previous work

[1] In Däubler et al. (2012), based on a sample of 15 manifestos, the authors noted that around 7.7% of sentences encode multiple topics.

(Biessmann, 2016; Glavaš et al., 2017), we consider the sentence-level classification to be a multi-class single-label problem. We use the segmented text when available (especially for evaluation), and complete sentences otherwise.

A manifesto as a whole can be positioned on the left–right spectrum based on the proportion of topics discussed. We use the RILE score, which is defined as the difference between the count of sentences discussing left- and right-leaning topics (Budge and Laver, 1992):

$$\text{RILE} = \sum_{r \in R} \text{per}_r - \sum_{l \in L} \text{per}_l \quad (1)$$

where R and L denote right and left political themes (see Figure 5), and per_t denotes the share of each topic t as given in Table 5, per document. Note that the RILE score is provided for almost all the manifestos in the CMP dataset, but the sentence-level annotations are provided only for a subset of manifestos. That is, in some cases, the underlying annotations that the RILE score calculation was based on is often not available for a given manifesto.

4 Proposed Approach

We propose a joint sentence–document model to classify manifesto sentences into one out of 57 categories and also quantify the document-level RILE score. The joint formulation is employed not only to capture the task inter-dependencies, but also to use annotations at different levels of granularity (sentence and document) effectively — a RILE score is available for 948 manifestos from 13 countries, whereas sentence-level annotations are available only for 235 manifestos. We use a hierarchical neural network to model the sentence-level classification and document-level regression tasks. The proposed architecture is given in Figure 1. Since the text across countries is multi-lingual in nature, we use multi-lingual embeddings to represent words (e_w) (Ammar et al., 2016). We refer to the total set of manifestos available for training as D, and the subset which is annotated with sentence-level labels as D_s. We denote each manifesto as d, which has l_d sentences $s_1, s_2, ..., s_{l_d}$. We also use i to index documents (i=d) wherever necessary to avoid ambiguity in differentiating from sentence-level variables.

4.1 Sentence-level Model

We represent each sentence using the average embedding of its constituent words, $\boldsymbol{s}_j = \frac{1}{|s_j|} \sum_{w \in s_j} e_w$. The average embedding representation is given as input to a hidden layer with rectified linear activation units (ReLU) to get the hidden representation. Finally, the predictions are obtained using a softmax layer, which takes the hidden representation as input and gives the probability of 57 classes as output, denoted $\hat{\mathbf{y}}_{ij}$. We use the cross-entropy loss function for the sentence-level model. For sentences in D_s, with ground truth labels $\mathbf{y}_{ij}$ (using a one-hot encoding), the loss function is given as follows:

$$\mathcal{L}_S = -\frac{1}{|D_s|} \sum_{i=1}^{|D_s|} \sum_{j=1}^{l_d} \sum_{k=1}^{K} y_{ijk} \log \hat{y}_{ijk} \quad (2)$$

4.2 Joint Sentence–Document Model

Using the hierarchical neural network, we model the sentence-level classification and document-level regression tasks together. In the joint model, we use an unrolled (time-distributed) neural network model for the sentences in a manifesto (d). Here, the model minimizes cross-entropy loss for sentences over each temporal layer ($j = 1 \ldots l_d$). We use average-pooling with the concatenated hidden representations ($\mathbf{h}_{ij}$) and predicted output distributions ($\hat{\mathbf{y}}_{ij}$) of individual sentences, to represent a document,[2] i.e., $\mathbf{r}_d = \frac{1}{|l_d|} \sum_{j \in d} \begin{bmatrix} \hat{\mathbf{y}}_{ij} \\ \mathbf{h}_{ij} \end{bmatrix}$.

The range of RILE is $[-100, 100]$, which we scale to the range $[-1, 1]$. Hence we use a final $\tanh$ layer, with $\hat{z}_i = \mathbf{w}_r^\top \mathbf{h}_d + b$, where $\mathbf{h}_d = \text{ReLU}(W_d^\top \mathbf{r}_d)$. Since it is a regression task, we minimize the mean-squared error loss function between the predicted $\hat{z}_i$ and actual RILE score z_i,

$$\mathcal{L}_D = \frac{1}{|D|} \sum_{i=1}^{|D|} ||\hat{z}_i - z_i||_2^2 \quad (3)$$

Overall, the loss function for the joint model, combining Equations 2 and 3, is:

$$\alpha \mathcal{L}_S + (1 - \alpha) \mathcal{L}_D \quad (4)$$

where $0 \leq \alpha \leq 1$ is a hyperparameter which is tuned on a development set.

We evaluate both cascaded and joint training for this objective function:

[2] We observed that the concatenated representation performed better than using either hidden representation or output distribution.

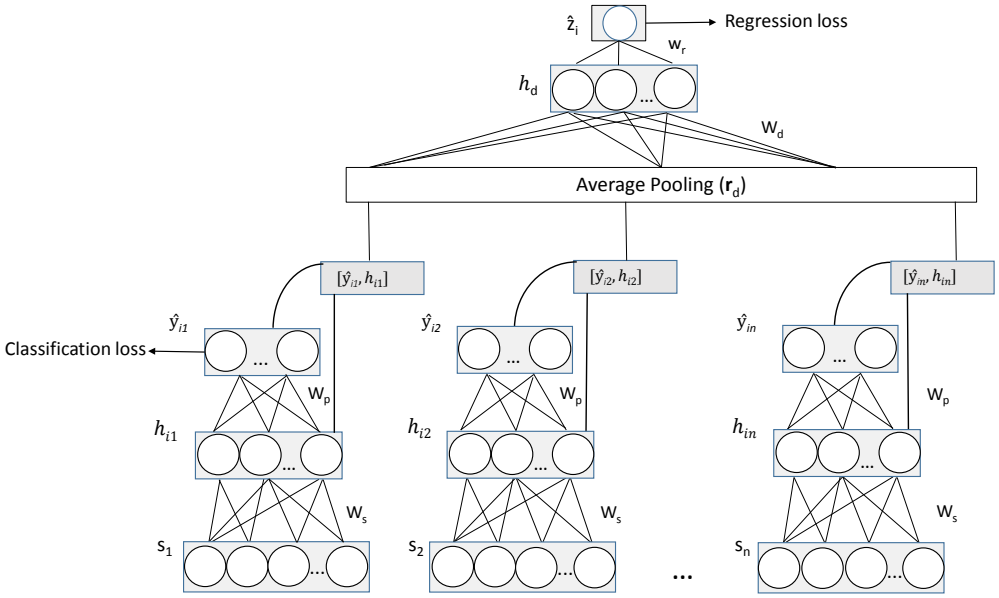


Figure 1: Hierarchical Neural Network for Joint Sentence–Document Analysis. s_1, s_2,...s_n are input sentences (n=l_d), W_s and W_p are shared across unrolled sentences. $\hat{\mathbf{y}}_{ij}$ denotes 57 classes and $\hat{z}_i$ denotes the estimated RILE score

Cascaded Training: The sentence-level model is trained using D_s, to minimize $\mathcal{L}_S$ in Equation 2, and the pre-trained sentence-level model is used to obtain document-level representation $\mathbf{r_d}$ for all the manifestos in the training set D. Then the document-level regression task is trained to minimize $\mathcal{L}_D$ from Equation 3. Here, the sentence-level model parameters are fixed when the document-level regression model is trained using $\mathbf{r_d}$.

Joint Training: The entire network is updated by minimizing the joint loss function from Equation 4. As in cascaded training, the sentence-level model is pre-trained using labeled sentences. Here the sentence-level model uses both labeled and unlabeled data.

We use the Adam optimizer (Kingma and Ba, 2014) for parameter estimation. The proposed architecture evaluates the effectiveness of posing sentence-level topic classification as a precursor to perform document-level RILE prediction, rather than learning a model directly. We also study the effect of the quantity of annotated text at both the sentence- and document-level for the RILE prediction task.

5 Experiments

5.1 Setting

As mentioned earlier, we use manifestos collected and annotated by political scientists as part of CMP. In this work, we used 948 manifestos from 13 countries, which are written in 6 different languages — Danish (Denmark), English (Australia, Ireland, New Zealand, South Africa, United Kingdom, United States), Finnish (Finland), French (France), German (Austria, Germany, Switzerland), and Italian (Italy). Out of the 948 manifestos, 235 are annotated with sentence level labels (from Table 5). We have RILE scores for all the 948 manifestos. Statistics about number of annotated documents and sentences across languages are given in Table 1. Class distribution based on average percentage of sentences coded under each class is given in Figure 2. Top-3 frequent set of classes include 000 (above 8%) , 504 (6-8%) and 305 & 503 (4-6%); and 26 classes occur 0-1%.

We use off-the-shelf pre-trained multi-lingual word embeddings[3] to represent words. We empirically chose embeddings trained using translation invariance approach (Ammar et al., 2016), with size 512 for our work. The neural network model has a single hidden layer for all the sentence and document-level approaches.

[3] http://128.2.220.95/multilingual

Lang.	# Docs (Ann.)	# Sents (Ann.)
Danish	175 (36)	32161 (8762)
English	312 (94)	227769 (73682)
Finnish	97 (16)	18717 (8503)
French	53 (10)	24596 (5559)
German	216 (65)	146605 (79507)
Italian	95 (14)	40010 (4918)
Total	948 (235)	489858 (180931)

Table 1: Statistics of dataset, 'Ann.' refers to annotated at sentence level.

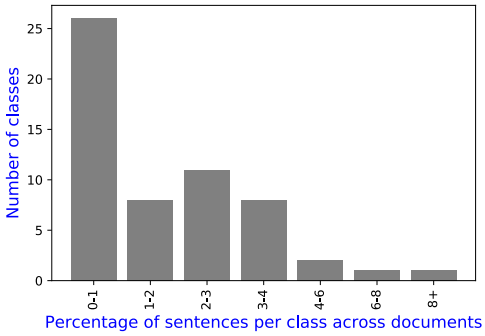


Figure 2: Class distribution based on average percentage of sentences coded under each class

5.2 Sentence-Level Classification

We first compare traditional bag-of-words discrete representation with distributed neural representation for words for fine-grained thematic classification, under mono-lingual training setting (*Mono-lingual).* Hence we compare the following approaches.

Bag-of-words (*BoW-LR,* BoW-NN): We use TF-IDF representation for sentences and build a model for each language separately. We use Logistic Regression classifier (Biessmann, 2016), which is referred as *BoW-LR.* We also use Neural Network classifier, which we refer to as *BoW-NN.*

Language-wise average embedding (*AE-NN$_m$):* We build a neural network classifier per language, with average multi-lingual neural embedding as sentence representation.

Since distributed representation allows to leverage text across languages, we evaluate the following approaches with combined training sentences across languages (*Cross-lingual).*

Convolutional Neural Network (*CNN):* CNN was shown to be effective for cross-lingual manifesto text coarse-level topic classification (Glavaš et al., 2017). So, we evaluate CNN with a similar architecture — single convolution layer (32 filters with window size 3), followed by single max pooling layer and finally a softmax layer. We use multi-lingual neural embeddings to represent words.

Combined average embedding (*AE-NN$_c$):* We build a neural network classifier with training instances combined across languages, with average neural embedding as sentence representation. This is our proposed approach for sentence-level model.

Commonly for all empirical evaluations, we compute micro-averaged performance with 80-20% train-test ratio across 10 runs with random split (at document level), where the 80% split also contains sentence level annotated documents proportionally. Optimal model parameters we found for the proposed model (Figure 1) are $|\mathbf{h}_{ij}| = 300$ (for sentences), $|\mathbf{h}_d| = 10$. We compute F-score[4] to evaluate sentence classification performance. Sentence classification performance is given in Table 2. Under mono-lingual setting (Table 2), using word embeddings did not provide better performance compared to bag-of-words.

Under cross-lingual setting, *AE-NN$_c$* is the sentence-level neural network model. We use *AE-NN$_c$* in the *cascaded training* for obtaining document-level RILE prediction. Note that in *cascaded training*, sentence and document-level models are trained separately in a cascaded fashion. Joint-training results where the sentence model is trained in a semi-supervised way together with document-level regression task is referred to as *JT$_s$*. We set α=0.4 (in equation 4) empirically which gave the best score for both sentence and document-level tasks. We observed a trade-off in performance with different α, with lesser α (0.1), document-level correlation increases (to 0.52) while sentence-level F-score decreases (to 0.33). Higher value of α (0.9) gives performance closer to cascaded training. *JT$_s$* has a comparable performance with *AE-NN$_c$*. The proposed approach (joint-training) does not provide any improvement for the sentence classification task.

[4]Harmonic mean of precision and recall, https://en.wikipedia.org/wiki/F1_score

	Mono-lingual			Cross-lingual		
Lang.	*BoW-LR*	*BoW-NN*	*AE-NN$_m$*	*CNN*	*AE-NN$_c$*	*JT$_s$*
da	0.29	**0.35**	0.24	0.30	0.28	0.30
en	0.36	0.38	**0.42**	0.40	**0.42**	0.41
fi	0.21	0.29	0.26	**0.30**	0.27	0.26
fr	0.28	0.36	0.24	0.36	0.37	**0.38**
de	0.30	0.31	0.31	0.31	0.31	**0.33**
it	0.32	**0.33**	0.25	0.30	0.32	0.26
Avg.	0.32	0.34	0.35	0.34	**0.36**	0.35

Table 2: Micro-Averaged F-measure for sentence classification. Best scores are given in bold.

5.3 Document-Level Regression

For the document-level regression task, the following are baseline approaches. Note that we use $\tanh$ output for all the models, since the range of re-scaled RILE is from -1 to +1.

Bag-of-words (*BoW-NN$_d$):* We use TF-IDF representation for documents and build a neural network model for each language.

Average embedding (*AE-NN$_d$):* We use average embedding of words as document representation to build a neural network model.

Bag-of-Centroids (*BoC):* Here the word embeddings are clustered into K different clusters using K-Means clustering algorithm, and words (1-gram) in each document are assigned to clusters based on its euclidean-distance (dist) to cluster-centroids (C) (Lebret and Collobert, 2014),

$$\text{cluster}(w) = \underset{k}{\text{argmin}}\, \text{dist}(C_k, w).$$

Finally, each document is represented by the distribution of words mapped to different clusters ($1 \times K$ vector). We use a neural network regression model with bag-of-centroids representation. Results with K=1000, which performed best is given in Table 3.

Sentence-level model and RILE formulation (*AE-NN$_c^{rile}$*): Here the predictions of sentence-level model (*AE-NN$_c$)* are used directly with RILE formulation (equation (1)) to derive RILE score for manifestos.

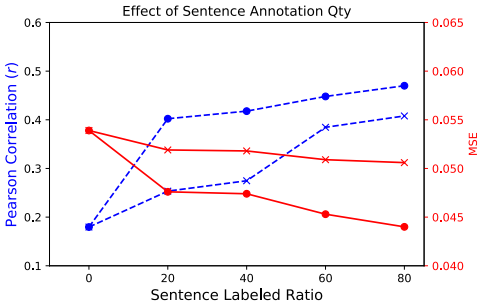
Cross-lingual scaling (*CLS):* This is a recent unsupervised approach for cross-lingual political speech text positioning task (Glavas et al., 2017). Authors use average word-embeddings weighed by TF-IDF score to represent documents.[5] Then a graph is constructed using pair-wise distance of documents. Given two pivots texts for extreme left and right positions [-1, +1], label propagation approach is used to quantify other documents in the graph.

Approach	MSE(↓)	r(↑)
BoW-NN$_d$	0.054	0.23
AE-NN$_d$	0.057	0.14
BoC	0.052	0.33
AE-NN$_c^{rile}$	0.060	0.35
CLS	–	0.24
Cas$_d$	0.050	0.41
JT$_d$	**0.044**	**0.47**

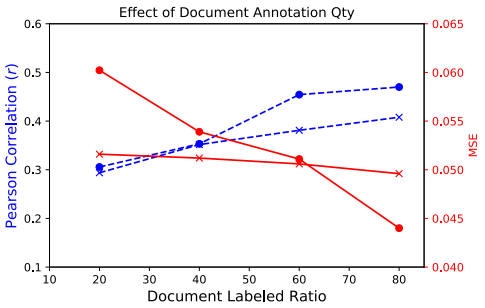
Table 3: RILE score prediction performance. Best scores are given in bold (higher is better for r, and lower is better for MSE).

RILE score regression performance results are given in Table 3. Other than *BoW-NN$_d$* all other approaches are cross-lingual. We evaluate document-level performance using mean-squared-error (MSE) and Pearson correlation (r). Since *CLS* solves it as a classification problem, MSE is not applicable. The proposed approach's performance, using cascaded training is referred to as *Cas$_d$* and jointly trained model is referred to as *JT$_d$*. Overall the jointly trained model performs best for document-level task, with a comparable performance at sentence-level task.

[5] We use this aggregate representation since it was shown to be better than word alignment and scoring approach (Glavas et al., 2017)



(a) Fixing 80% training documents with RILE score, ratio of documents with sentence-level annotations is varied.



(b) Fixing 80% training documents with sentence-level annotations, ratio of documents with RILE score is varied.

Figure 3: Study with Quantity of Annotation. In 3(a) and 3(b) cross(×) denotes Cas_d and circle(○) denotes JT_d

5.4 Quantity of Annotation

We measure the importance of annotated text at sentence and document-level for RILE score regression task. We vary the percentage of labeled data, while keeping the test sample size at 20% as before. In the first setting, we keep the training ratio of documents at 80%, within that 80% we increase the proportion of documents with sentence-level annotations — from 0 (document average embedding setting, *AE-NN$_d$)* to 80%. Results are given in Figure 3a. Similarly, in the other setting, we keep the training set with 80% sentence-level annotated documents (which is ∼20% of the total data), and add documents (with only RILE score), increasing the training set from 20 to 80%. Results of this study are given in Figure 3b. We observed that, jointly-trained model uses sentence-level annotations more effectively than cascaded approach (Figure 3a) — even with less sentence-level annotations. Also, with less document-level signal (up to 40%) for training, both the approaches perform similarly (r). As the training ratio increases, joint-training leverages both sentence and document-level signals effectively.

Approach	MSE	r
stack	0.045 (0.001 ↓)	0.49 (0.02 ↑)
non-linear stack	0.048 (0.004 ↓)	0.48 (0.01 ↑)

Table 4: RILE score prediction performance with *country* information. Difference compared to JT_d is given within paranthesis. ↑ – improvement, ↓ – decrease in performance

5.5 Use of Country Information

Since the definition of left–right varies between countries, we study the influence of *country* information in the proposed model with *joint-training*. We use two ways to incorporate country information (Hoang et al., 2016): (a) *stack* — one-hot encoding (13 countries, 1×13 vector) of each manifesto's *country* is concatenated with hidden representation of the document ($\mathbf{r}_d$ in Figure 1) (b) *non-linear stack* — one-hot-encoded country vector is passed through a hidden layer with $\tanh$ non-linear activation and concatenated with $\mathbf{r}_d$. With both the models we observed mild improvement in correlation (given in Table 4).

6 Conclusion and Future Work

In this work we evaluated the utility of a joint sentence–document model for sentence-level thematic classification and document-level RILE score regression tasks. Our observations are as follows: (a) joint model performs better than state-of-art approaches for document-level regression task (b) joint-training leverages sentence-level annotations more effectively than cascaded approach for RILE score regression task, with no gains for sentence classification task. There are many extensions possible to the current work. First is to handle class imbalance in the dataset with a cost-sensitive objective function. Secondly, CNN gave a comparable performance with Neural Network, which motivates the need to evaluate an end-end sequential architecture to obtain sentence and document embeddings. Off-the-shelf embeddings leads to out-of-vocabulary scenarios. It could be beneficial to adapt word-embeddings with manifesto corpus. Finally, background information such as country can be leveraged more effectively.

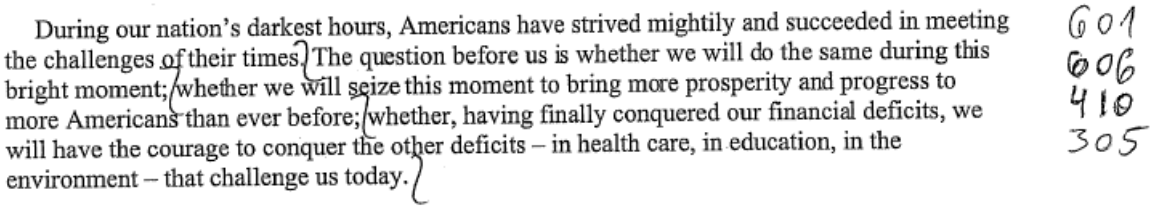


Figure 4: Manifesto snippet for Democratic Party of USA, 2000 — $\int$ denotes sentence segment. See Table 5 for code description.

CMP Coding Scheme	
• Domain 1. External Relations	411 Technology and Infrastructure: Positive
101 Foreign Special Relationships: Positive	412 Controlled Economy
102 Foreign Special Relationships: Negative	413 Nationalisation
103 Anti-Imperialism	414 Economic Orthodoxy
104 Military: Positive	415 Marxist Analysis
105 Military: Negative	416 Anti-Growth Economy: Positive
106 Peace	
107 Internationalism: Positive	• Domain 5: Welfare and Quality of Life
108 European Community/Union: Positive	501 Environmental Protection
109 Internationalism: Negative	502 Culture: Positive
110 European Community/Union: Negative	503 Equality: Positive
	504 Welfare State Expansion
• Domain 2: Freedom and Democracy	505 Welfare State Limitation
201 Freedom and Human Rights	506 Education Expansion
202 Democracy	507 Education Limitation
203 Constitutionalism: Positive	
204 Constitutionalism: Negative	• Domain 6: Fabric of Society
	601 National Way of Life: Positive
• Domain 3: Political System	602 National Way of Life: Negative
301 Decentralisation	603 Traditional Morality: Positive
302 Centralisation	604 Traditional Morality: Negative
303 Governmental and Administrative Efficiency	605 Law and Order: Positive
304 Political Corruption	606 Civic Mindedness: Positive
305 Political Authority	607 Multiculturalism: Positive
	608 Multiculturalism: Negative
• Domain 4: Economy	
401 Free Market Economy	• Domain 7: Social Groups
402 Incentives: Positive	701 Labour Groups: Positive
403 Market Regulation	702 Labour Groups: Negative
404 Economic Planning	703 Agriculture and Farmers: Positive
405 Corporatism/Mixed Economy	704 Middle Class and Professional Groups
406 Protectionism: Positive	705 Underprivileged Minority Groups
407 Protectionism: Negative	706 Non-economic Demographic Groups
408 Economic Goals	
409 Keynesian Demand Management	
410 Economic Growth: Positive	000 No meaningful category applies

Table 5: Comparative Manifesto Project — 57 policy themes. *Left* topics are given in *red* and *right* topics are given in *blue* and the *rest* are considered neutral

References

Leman Akoglu. 2014. Quantifying political polarity based on bipartite opinion networks. In *AAAI*.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925* .

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. In *ACL*.

Kenneth Benoit and Thomas Däubler. 2014. Putting text in context: How to estimate better left-right positions by scaling party manifesto data using item response theory. In *Mapping Policy Preferences from Texts Conference*.

Felix Biessmann. 2016. Automating political bias prediction. In *arXiv:1608.02195*.

B. Bruinsma and K. Gemenis. 2017. Validating Wordscores. *ArXiv e-prints* .

Ian Budge and Michael Laver. 1992. *Party policy and government coalitions*. St. Martin's Press New York.

Thomas Däubler, Kenneth Benoit, Slava Mikhaylov, and Michael Laver. 2012. Natural sentences as valid units for coded political texts. *British Journal of Political Science* .

Zhuoye Ding, Qi Zhang, and Xuanjing Huang. 2011. Keyphrase extraction from online news using binary integer programming. In *IJCNLP*.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. Cross-lingual classification of topics in political texts. In *ACL WS*.

Goran Glavas, 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* .

Frederik Hjorth, Robert Klemmensen, Sara Hobolt, Martin Ejnar Hansen, and Peter Kurrild-Klitgaard. 2015. Computers, coders, and voters: Comparing automated methods for estimating party positions. *Research & Politics* .

Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. 2016. Incorporating side information into recurrent neural network language models. In *NAACL-HLT*.

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

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.

Rémi Lebret and Ronan Collobert. 2014. N-gram-based low-dimensional representation for document classification. *arXiv preprint arXiv:1412.6277* .

Will Lowe, Kenneth Benoit, Slava Mikhaylov, and Michael Laver. 2011. Scaling policy preferences from coded political texts. In *Legislative studies quarterly*. Wiley Online Library.

Christopher Lucas, 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.

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

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*.

Ludovic Rheault. 2016. Expressions of anxiety in political texts. In *NLP+ CSS 2016*. page 92.

Jonathan B Slapin and Sven-Oliver Proksch. 2008. A scaling model for estimating time-series party positions from texts. Wiley Online Library, volume 52, pages 705–722.

Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective stance classification of posts in online debate forums. In *ACL*.

Suzan Verberne, Eva D'hondt, Antal van den Bosch, and Maarten Marx. 2014. Automatic thematic classification of election manifestos. In *Information Processing & Management*.

Andrea Volkens, Onawa Lacewell, and Sven Regel Henrike Schultzeand Annika Werner. Pola Lehmann. 2011. The manifesto data collection. manifesto project (mrg/cmp/marpor). In *Wissenschaftszentrum Berlin fur Sozialforschung (WZB)*.

Daniel Xiaodan Zhou, Paul Resnick, and Qiaozhu Mei. 2011. Classifying the political leaning of news articles and users from user votes. In *ICWSM*.

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