A Simple Regularization-based Algorithm for Learning Cross-Domain Word Embeddings

Wei Yang
University of Waterloo
w85yang@uwaterloo.ca

Wei Lu
Singapore University of Technology and Design
luwei@sutd.edu.sg

Vincent W. Zheng
Advanced Digital Sciences Center
vincent.zheng@adsc.com.sg

Abstract

Learning word embeddings has received a significant amount of attention recently. Often, word embeddings are learned in an unsupervised manner from a large collection of text. The genre of the text typically plays an important role in the effectiveness of the resulting embeddings. How to effectively train word embedding models using data from different domains remains a problem that is underexplored. In this paper, we present a simple yet effective method for learning word embeddings based on text from different domains. We demonstrate the effectiveness of our approach through extensive experiments on various down-stream NLP tasks.

1 Introduction

Recently, the learning of distributed representations for natural language words (or word embeddings) has received a significant amount of attention (Mnih and Hinton, 2007; Turian et al., 2010; Mikolov et al., 2013a,b,c; Pennington et al., 2014). Such representations were shown to be able to capture syntactic and semantic level information associated with words (Mikolov et al., 2013a). Word embeddings were shown effective in tasks such as named entity recognition (Sienčnik, 2015), sentiment analysis (Li and Lu, 2017) and syntactic parsing (Durrett and Klein, 2015). One common assumption made by most of the embedding methods is that, the text corpus is from one single domain; e.g., articles from bioinformatics. However, in practice, there are often text corpora from multiple domains; e.g., we may have text collections from broadcast news or Web blogs, whose words are not necessarily limited to bioinformatics. Can these corpora from different domains help learn better word embeddings, so as to improve the downstream NLP applications in a target domain like bioinformatics? Our answer is yes, because despite the domain differences, these additional domains do introduce more text data converying useful information (i.e., more words, more word co-occurrences), which can be helpful for consolidating the word embeddings in the target bioinformatics domain.

In this paper, we propose a simple and easy-to-implement approach for learning cross-domain word embeddings. Our model can be seen as a regularized *skip-gram* model (Mikolov et al., 2013a,b), where the source domain information is selectively incorporated for learning the target domain word embeddings in a principled manner.

2 Related Work

Learning a continuous representation for words has been studied for quite a while (Hinton et al., 1986). Many earlier word embedding methods employed the computationally expensive neural network architectures (Collobert and Weston, 2008; Mikolov et al., 2013c). Recently, an efficient method for learning word representations, namely the skip-gram model (Mikolov et al., 2013a,b) was proposed and implemented in the widely used word2vec toolkit. It tries to use the current word to predict the surrounding context words, where the prediction is defined over the embeddings of these words. As a result, it learns the word embeddings by maximizing the likelihood of predictions.

Domain adaptation is an important research topic (Pan et al., 2013), and it has been considered in many NLP tasks. For example, domain adaptation is studied for sentiment classification (Glorot et al., 2011) and parsing (McClosky et al., 2010), just to name a few. However, there is very

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2898–2904
Copenhagen, Denmark, September 7–11, 2017.

little work on domain adaptation for word embedding learning. One major reason preventing people from using text corpora from different domains for word embedding learning is the lack of guidance on which kind of information is worth learning from the source domain(s) for the target domain. In order to address this problem, some pioneering work has looked into this problem. For example, Bollegala et al. (2015) considered those frequent words in the source domain and the target domain as the "pivots". Then it tried to use the pivots to predict the surrounding "non-pivots", meanwhile ensuring the pivots to have the same embedding across two domains. Embeddings learned from such an approach were shown to be able to improve the performance on a cross-domain sentiment classification task. However, this model fails to learn embeddings for many words which are neither pivots nor non-pivots, which could be crucial for some downstream tasks such as named entity recognition.

3 Our Approach

Let us first state the objective of the skip-gram model (Mikolov et al., 2013a) as follows:

$$\mathcal{L}_{\mathcal{D}} = \sum_{(w,c)\in\mathcal{D}} \#(w,c)\Big(\log\sigma(\mathbf{w}\cdot\mathbf{c}) + \sum_{i=1}^{k}\mathbb{E}_{c'_i\sim P(w)}[\log\sigma(-\mathbf{w}\cdot\mathbf{c'}_i)]\Big) \quad (1)$$

where $\mathcal{D}$ refers to the complete text corpus from which we learn the word embeddings. The word w is the current word, c is the context word, and $\#(w,c)$ is the number of times they co-occur in $\mathcal{D}$. We use $\mathbf{w}$ and $\mathbf{c}$ to denote the vector representations for w and c, respectively. The function $\sigma(\cdot)$ is the sigmoid function. The word c'_i is a "negative sample" sampled from the distribution $P(w)$ – typically chosen as the unigram distribution $U(w)$ raised to the $3/4$rd power (Mikolov et al., 2013b).

In our approach, we first learn for each word w an embedding $\mathbf{w}_s$ from the source domain $\mathcal{D}_s$. Next we learn the target domain embeddings as follows:

$$\mathcal{L}'_{\mathcal{D}_t} = \mathcal{L}_{\mathcal{D}_t} + \sum_{w\in\mathcal{D}_t\cap\mathcal{D}_s} \alpha_w\cdot||\mathbf{w}_t-\mathbf{w}_s||^2 \quad (2)$$

where $\mathcal{D}_t$ refers to the target domain, and $\mathbf{w}_t$ is the target domain representation for w. Such an regularized objective can still be optimized using standard stochastic gradient descent. Note that in the above formula, the regularization term only considers words that appear in both source and target domain, ignoring words that only appear in either the source or the target domain only.

Our approach is inspired by the recent regularization-based domain adaptation framework (Lu et al., 2016). Here, α_w measures the amount of transfer across the two domains when learning the representation for word w. If it is large, it means we require the embeddings of word w in the two domains to be similar. We define α_w as follows:

$$\alpha_w = \sigma(\lambda\cdot\phi(w)) \quad (3)$$

where λ is a hyper-parameter to decide the scaling factor of the significance function $\phi(\cdot)$, which allows the user to control the degree of "knowledge transfer" from source domain to target domain.

How do we define the significance function $\phi(w)$ that controls the amount of transfer for the word w? We first define the frequency of the word w in the dataset $\mathcal{D}$ as $f_{\mathcal{D}}(w)$, the number of times the word w appears in the domain $\mathcal{D}$. Based on this we can define the *normalized* frequency for the word w as follows:

$$\mathcal{F}_{\mathcal{D}}(w) = \frac{f_{\mathcal{D}}(w)}{\max_{w'\in\mathcal{D}_k} f_{\mathcal{D}}(w')} \quad (4)$$

where $\mathcal{D}_k\subset\mathcal{D}$ consists of all except for the top k most frequent words from $\mathcal{D}$[1].

We define the function $\phi(\cdot)$ based on the following metric that is motivated by the well-known Sørensen-Dice coefficient (Sørensen, 1948; Dice, 1945) commonly used for measuring similarities:

$$\phi(w) = \frac{2\cdot\mathcal{F}_{\mathcal{D}_s}(w)\cdot\mathcal{F}_{\mathcal{D}_t}(w)}{\mathcal{F}_{\mathcal{D}_s}(w)+\mathcal{F}_{\mathcal{D}_t}(w)} \quad (5)$$

Why does such a definition make sense? We note that the value of $\phi(w)$ would be high only if both both $\mathcal{F}_{\mathcal{D}_s}(w)$ and $\mathcal{F}_{\mathcal{D}_t}(w)$ are high – in this case the word w is a frequent word across different domains. Intuitively, these are likely those words whose semantics do not change across the two domains, and we should be confident about making their embeddings similar in the two domains. On the other hand, domain-specific words

[1] In all our experiments, we empirically set k to 20.

	Enwik9	PubMed	Gigaword (EN)	Yelp	IMDB	Tweets (EN)	Tweets (ES)	Eswiki
# tokens	124.3M	124.9M	135.6M	38.9M	29.0M	162.8M	69.4M	102.8M
# sents	–	5,000,000	5,400,000	2,376,079	1,230,465	16,185,356	6,785,697	3,684,670

Table 1: Statistics for datasets used for embedding learning in all experiments.

tend to be more frequent in one domain than the other. In this case, the resulting $\phi(w)$ will also have a lower score, indicating a smaller amount of transfer across the two domains. While other user-defined significance functions are also possible, in this work we simply adopt such a function based on the above simple observations. We will validate our assumptions with experiments in the next section.

4 Experiments

We present extensive evaluations to assess the effectiveness of our approach. Following recent advice by Nayak et al. (2016) and Faruqui et al. (2016), to assess the quality of the learned word embeddings, we considered employing the learned word embeddings as continuous features in several down-stream NLP tasks, including entity recognition, sentiment classification, and targeted sentiment analysis.

We have used various datasets from different domains for learning cross-domain word embeddings under different tasks. We list the data statistics in Table 1.

4.1 Baseline Methods

We consider the following baseline methods when assessing the effectiveness of our approach.

- DISCRETE: only discrete features (such as bag of words, POS tags, word n-grams and POS tag n-grams, depending on the actual down-stream task) were considered. All following systems include both these base features and the respective additional features.
- SOURCE: we train word embeddings from the source domain as additional features.
- TARGET: we train word embeddings from the target domain as additional features.
- ALL: we combined the data from two domains to form a single dataset for learning word embeddings as additional features.
- CONCAT: we simply concatenate the learned embeddings from both source and target domains as additional features.

Method	GENIA			ACE		
	P	R	F_1	P	R	F_1
DISCRETE	71.1	63.9	67.3	64.5	52.3	57.7
SOURCE	71.1	62.3	66.4	63.5	57.3	60.3
TARGET	71.6	64.5	67.9	63.3	57.1	60.0
ALL	71.2	61.8	66.1	**64.6**	57.2	60.7
CONCAT	71.5	64.1	67.6	63.5	57.7	60.5
DARep	71.4	61.5	66.1	62.4	54.5	58.2
This work	**72.4**	**65.4**	**68.7**	64.5	**58.9**	**61.6**

Table 2: Results on entity recognition.

- DARep: we use the previous approach of Bollegala et al. (2015) for learning cross-lingual word representations as additional features.

4.2 Entity Recognition

Our first experiment was conducted on entity recognition (Tjong Kim Sang and De Meulder, 2003; Florian et al., 2004), where the task is to extract semantically meaning entities and their mentions from the text.

For this task, we built a standard entity recognition model using conditional random fields (Lafferty et al., 2001). We used the standard features which are commonly used for different methods, including word unigrams and bigrams, bag-of-words features, POS tag window features, POS tag unigrams and bigram features. We conducted two sets of experiments on two different datasets. The first dataset is the GENIA dataset (Ohta et al., 2002), a popular dataset used in bioinformatics, and the second is the ACE-2005 dataset (Walker et al., 2006), which is a standard dataset used for various information extraction tasks.

For the GENIA dataset which consists of 10,946 sentences, we used Enwik9 as the source domain and PubMed as the target domain for learning word embeddings. We set the dimension of word representations as 50.

For the experiments on ACE, we selected the BN subset of ACE2005, which consists of 4,460 CNN headline news and share a similar domain with Gigaword. We used Enwik9 as the source domain and Gigaword as the target domain. We followed a procedure similar to GENIA for experiments.

To tune our hyperparameter λ, we first split the last 10% of the training set as the development

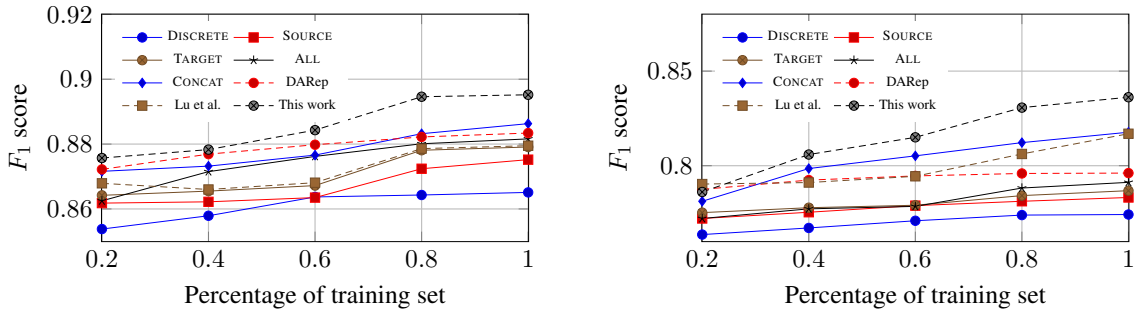

Figure 1: Results on sentiment classification. Left: Yelp (source) to IMDB (target). Right: IMDB (source) to Yelp (target).

portion. We then trained a model using the remaining 90% as the training portion and used the development portion for development of the hyperparameter λ. After development, we re-trained the models using the original training set[2].

We report the results in Table 2. From the results we can observe that the embeddings learned using our algorithm can lead to improved performance when used in this particular down-stream NLP task. We note that in such a task, many entities consist of domain-specific terms, therefore learning good representations for such words can be crucial. As we have discussed earlier, our regularization method enables our model to differentiate domain-specific words from words which are more general in the learning process. We believe this mechanism can lead to improved learning of representations for both types of words.

4.3 Sentiment Classification

The second task we consider is sentiment classification, which is essentially a text classification task, where the goal is to assign each text document a class label indicating its sentiment polarity (Pang et al., 2002; Liu, 2012).

This is also the only task presented in the previous DARep work by Bollegala et al. (2015). As such, we largely followed Bollegala et al. (2015) for experiments. Instead of using the dataset they used which only consists of 2,000 reviews, we considered two much larger datasets – IMDB and Yelp 2014 – for such a task, which was previously used in a sentiment classification task (Tang et al., 2015). IMDB dataset (Diao et al., 2014) is crawled from the movie review site IMDB[3] which consists of 84,919 reviews. Yelp 2014 dataset consists of 231,163 online reviews provided by the Yelp Dataset Challenge[4].

Following Bollegala et al. (2015), for this task we simply learned the word embeddings from the training portion of the review datasets themselves only. No external data was used for learning word embeddings. As Bollegala et al. (2015) only evaluated on a small dataset in their paper for such a task, to understand the effect of varying the amount of training data, we also tried to train our model on datasets with different sizes. We conducted two sets of experiments: we first used the Yelp dataset as the source domain and IMDB as the target domain, and then we switched these two datasets in our second set of experiments. Figure 1 shows the F_1 measures for different word embeddings when different amounts of training data were used. We also compared with the previous approach for domain adaptation (Lu et al., 2016) which only employs discrete features. We can observe that when the dataset becomes large, our learned word embeddings are shown to be more effective than all other approaches. When the complete training set is used, our model significantly outperforms DARep ($p < 0.05$ for both directions with bootstrap resampling test (Koehn, 2004)). DARep appears to be effective when the training dataset is small. However, as the training set size increases, there is no significant improvement for such an approach. As we can also observe from the figure, our approach consistently gives better results than baseline approaches (except for the second experiment when 20% of the data was used). Furthermore, when the amount of training data increases, the differences between our approach and other approaches generally become larger.

Such experiments show that our model works

[2]We selected the optimal value for the hyper-parameter λ from the set $\lambda \in \{0.1, 1, 5, 10, 20, 30, 50\}$ for all experiments in this paper.

[3]http://www.imdb.com

[4]https://www.yelp.com/dataset_challenge

Model	English			Spanish		
	$P.$	$R.$	F_1	$P.$	$R.$	F_1
DISCRETE	44.8	37.0	40.5	46.0	39.8	42.7
SOURCE	44.1	36.3	39.8	46.1	40.5	43.1
TARGET	46.5	39.1	42.5	46.5	40.8	43.4
ALL	45.4	37.0	40.8	46.4	40.7	43.3
CONCAT	46.7	39.3	42.7	**46.6**	41.0	43.6
DAREp	46.2	39.8	42.8	46.2	40.9	43.4
This work	**46.9**	**39.9**	**43.1**	**46.6**	**41.4**	**43.9**

Table 3: Results on targeted sentiment analysis.

well when different amounts of data are available, and our approach appears to be more competitive when a large amount of data is available.

4.4 Targeted Sentiment Analysis

We also conducted experiments on targeted sentiment analysis (Mitchell et al., 2013) – the task of jointly recognizing entities and their sentiment information. We used the state-of-the-art system for targeted sentiment analysis by Li and Lu (2017) whose code is publicly available [5], and used the data from (Mitchell et al., 2013) which consists of 7,105 Spanish tweets and 2,350 English tweets, with named entities and their sentiment information annotated. Note that the model of Li and Lu (2017) is a structured prediction model that involves latent variables. The experiments here therefore allow us to assess the effectiveness of our approach on such a setup involving latent variables. We follow Li and Lu (2017) and report precision ($P.$), recall ($R.$) and F1-measure (F_1) for such a targeted sentiment analysis task, where the prediction is regarded as correct if and only if both the entity's boundary and its sentiment information are correct. Also, unlike previous experiments, which are conducted on English only, these experiments additionally allow us to assess our approach's effectiveness when a different language other than English is considered.

For the English task, we used Enwik9 as the source domain for learning word embeddings, and our crawled English tweets as the target domain. For the Spanish task, we used Eswiki as the source domain, and we also crawled Spanish tweets as the target domain. See Table 1 for the statistics. Similar to the experiments conducted for entity recognition, we split the first 80% of the data for training, the next 10% for development and the last 10% for evaluation. We tuned the hyper-parameter λ using the development set and re-trained the embeddings on the dataset combining the training and the development set, which are then used in final evaluations. Results are reported in Table 3, which show our approach is able to achieve the best results across two datasets in such a task, and outperforms DAREp ($p < 0.05$). Interestingly, the concatenation approach appears to be competitive in this task, especially for the Spanish dataset, which appears to be better than the DAREp approach. However, we note such an approach does not capture any information transfer across different domains in the learning process. In contrast, our approach learns embeddings for the target domain by capturing useful cross-domain information and therefore can lead to improved modeling of embeddings that are shown more helpful for this specific down-stream task.

[5] Available at http://statnlp.org/research/st/.

5 Conclusion and Future Work

In this paper, we presented a simple yet effective algorithm for learning cross-domain word embeddings. Motivated by the recent regularization-based domain adaptation framework (Lu et al., 2016), the algorithm performs learning by augmenting the skip-gram objective with a simple regularization term. Our work can be easily extended to multi-domain scenarios. The method is also flexible, allowing different user-defined metrics to be incorporated for defining the function controlling the amount of domain transfer.

Future work includes performing further investigations to better understand and to visualize what types of information has been transferred across domains and how such information influence different types of down-stream NLP tasks. It is also important to understand how such an approach will work on other types of models such as neural networks based NLP models. Our code is available at http://statnlp.org/research/lr/.

Acknowledgments

We thank all the reviewers for their useful feedback to the earlier draft of this paper. This work was done when the first author was visiting Singapore University of Technology and Design. We thank the support of Human-centered Cyber-physical Systems Programme at Advanced Digital Sciences Center from Singapores Agency for Science, Technology and Research (A*STAR). This work is supported by MOE Tier 1 grant SUTDT12015008.

References

Danushka Bollegala, Takanori Maehara, and Ken ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *Proc. of ACL-IJCNLP*, pages 730–740.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*, pages 160–167. ACM.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proc. of KDD*, pages 193–202.

Lee R Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Greg Durrett and Dan Klein. 2015. Neural crf parsing. *arXiv preprint arXiv:1507.03641*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proc. of NAACL/HLT*, pages 1–8.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proc. of ICML*, pages 513–520.

G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. 1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Distributed Representations, pages 77–109. MIT Press, Cambridge, MA, USA.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*, pages 388–395.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *Proc. of AAAI*, pages 3482–3489.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1).

Wei Lu, Hai Leong Chieu, and Jonathan Löfgren. 2016. A general regularization framework for domain adaptation. In *Proc. of EMNLP*, pages 950–954.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Proc. of NAACL-HLT*, pages 28–36.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proc. of NAACL-HLT*, pages 746–751.

Margaret Mitchell, Jacqueline Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proc. of EMNLP*, pages 1643–1654.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proc. of ICML*, pages 641–648.

Neha Nayak, Gabor Angeli, and Christopher D Manning. 2016. Evaluating word embeddings using a representative suite of practical tasks. *ACL 2016*, page 19.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proc. of the second international conference on Human Language Technology Research*, pages 82–86.

Sinno Jialin Pan, Zhiqiang Toh, and Jian Su. 2013. Transfer joint embedding for cross-domain named entity recognition. *ACM Transactions on Information Systems (TOIS)*, 31(2):7.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proc. of EMNLP*, pages 79–86.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*, volume 14, pages 1532–1543.

Scharolta Katharina Sienčnik. 2015. Adapting word2vec to named entity recognition. In *Proc. of NODALIDA*, 109, pages 239–243. Linköping University Electronic Press.

Thorvald Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proc. of ACL-IJCNLP*, pages 1014–1023.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proc. of HLT-NAACL*, pages 142–147.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.