Unsupervised Word and Dependency Path Embeddings for Aspect Term Extraction

Yichun Yin\textsuperscript{1}, Furu Wei\textsuperscript{2}, Li Dong\textsuperscript{3}, Kaimeng Xu\textsuperscript{1}, Ming Zhang\textsuperscript{1}\ast, Ming Zhou\textsuperscript{2}

\textsuperscript{1}School of EECS, Peking University
\textsuperscript{2}Microsoft Research
\textsuperscript{3}Institute for Language, Cognition and Computation, University of Edinburgh
\{yichun\textsubscript{yin},1300012834,mzhang\_cs\}@pku.edu.cn,\{fuwei,mingzhou\}@microsoft.com,li.dong@ed.ac.uk

Abstract

In this paper, we develop a novel approach to aspect term extraction based on unsupervised learning of distributed representations of words and dependency paths. The basic idea is to connect two words (\(w_1\) and \(w_2\)) with the dependency path (\(r\)) between them in the embedding space. Specifically, our method optimizes the objective \(w_1 + r \approx w_2\) in the low-dimensional space, where the multi-hop dependency paths are treated as a sequence of grammatical relations and modeled by a recurrent neural network. Then, we design the embedding features that consider linear context and dependency context information, for the conditional random field (CRF) based aspect term extraction. Experimental results on the SemEval datasets show that, (1) with only embedding features, we can achieve state-of-the-art results; (2) our embedding method which incorporates the syntactic information among words yields better performance than other representative ones in aspect term extraction.

1 Introduction

Aspect term extraction [Hu and Liu, 2004; Pontiki et al., 2014; 2015] aims to identify the aspect expressions which refer to the product’s or service’s properties (or attributes), from the review sentence. It is a fundamental step to obtain the fine-grained sentiment of specific aspects of a product, besides the coarse-grained overall sentiment. Until now, there have been two major approaches for aspect term extraction. The unsupervised (or rule based) methods [Qiu et al., 2011] rely on a set of manually defined opinion words as seeds and rules derived from syntactic parsing trees to iteratively extract aspect terms. The supervised methods [Jakob and Gurevych, 2010; Li et al., 2010; Chernyshievich, 2014; Toh and Wang, 2014; San Vicente et al., 2015] usually treat aspect term extraction as a sequence labeling problem, and conditional random field (CRF) has been the mainstream method in the aspect term extraction task of SemEval.

Representation learning has been introduced and achieved success in natural language processing (NLP) [Bengio et al., 2013], such as word embeddings [Mikolov et al., 2013b] and structured embeddings of knowledge bases [Bordes et al., 2011]. It learns distributed representations for text in different granularities, such as words, phrases and sentences, and reduces data sparsity compared with the conventional one-hot representation. The distributed representations have been reported to be useful in many NLP tasks [Turian et al., 2010; Collobert et al., 2011].

In this paper, we focus on representation learning for aspect term extraction under an unsupervised framework. Besides words, dependency paths, which have been shown to be important clues in aspect term extraction [Qiu et al., 2011], are also taken into consideration. Inspired by the representation learning of knowledge bases [Bordes et al., 2011; Neelakantan et al., 2015; Lin et al., 2015] that embeds both entities and relations into a low-dimensional space, we learn distributed representations of words and dependency paths from the text corpus. Specifically, the optimization objective is formalized as \(w_1 + r \approx w_2\). In the triple \((w_1, w_2, r)\), \(w_1\) and \(w_2\) are words, \(r\) is the corresponding dependency path consisting of a sequence of grammatical relations. The recurrent neural network [Mikolov et al., 2010] is used to learn the distributed representations of dependency paths. Furthermore, the word embeddings are enhanced by linear context information in a multi-task learning manner.

The learned embeddings of words and dependency paths are utilized as features in CRF for aspect term extraction. The embeddings are real values that are not necessarily in a bounded range [Turian et al., 2010]. We therefore firstly map the continuous embeddings into the discrete embeddings and make them more appropriate for the CRF model. Then, we construct the embedding features which include the target word embedding, linear context embedding and dependency context embedding for aspect term extraction. We conduct experiments on the SemEval datasets and obtain comparable performances with the top systems. To demonstrate the effectiveness of the proposed embedding method, we also compare our method with other state-of-the-art models. With the same feature settings, our approach achieves better results. Moreover, we perform a qualitative analysis to show the effectiveness of the learned word and dependency path embeddings.

The contributions of this paper are two-fold. First, we use the dependency path to link words in the embedding space for distributed representation learning of words and depen-
ence paths. By this method, the syntactic information is encoded in word embeddings and the multi-hop dependency path embeddings are learned explicitly. Second, we construct the CRF features only based on the derived embeddings for aspect term extraction, and achieve state-of-the-art results.

2 Related Work

There are a number of unsupervised methods for aspect term extraction. Hu and Liu [2004] mine aspect term based on a data mining approach called association rule mining. In addition, they use opinion words to extract infrequent aspect terms. Using relationships between opinion words and aspect terms, they use opinion words to extract infrequent aspect data mining approach called association rule mining. In addition, they use opinion words to extract infrequent aspect term extraction. Hu and Liu [2004] mine aspect term based on a multi-hop dependency path. Among these algorithms, the mainstream method is the CRF model. Li et al. [2010] propose a new machine learning framework on top of CRF to jointly extract opinion words, negative opinion words and aspect terms. In order to obtain a more domain independent extraction, Jakob and Gurevych [2010] train the CRF model on review sentences from different domains. Most of top systems in SemEval also rely on the CRF model, and hand-crafted features are constructed to boost performances.

The representation learning has been employed in NLP [Bengio et al., 2013]. Two successful applications are the word embedding [Mikolov et al., 2013b] and the knowledge embedding [Bordes et al., 2011]. The word embeddings are learned from raw textual data and are effective representations for word meanings. They have been proven useful in many NLP tasks [Turian et al., 2010; Guo et al., 2014]. The knowledge embedding method models the relation as a translation vector that connects the vectors of two entities. It optimizes an objective over all the facts in the knowledge bases, encoding the global information. The learned embeddings are useful for reasoning missing facts and facts extraction in knowledge bases.

As dependency paths contain rich linguistics information between words, recent works model it as a dense vector. Liu et al. [2015b] incorporate the vectors of grammatical relations (one-hop dependency paths) into the relation representation between two entities for relation classification task. Dependency-based word embedding model [Levy and Goldberg, 2014] encodes dependency information into word embeddings which is one similar work to ours. However, it implicitly encodes the dependency information, embedding the unit word + dependency path as the context vector. Besides, it only considers one-hop dependency paths and ignores multi-hop dependency paths. In this paper, we take the Dependency-based word embedding as a baseline method.

The approaches that learn path embeddings are investigated in knowledge bases. Neelakantan et al. [2015] provide a method that reasons about conjunctions of multi-hop relations for knowledge base completion. The implication of a path is composed by using a recurrent neural network. Lin et al. [2015] also learn the representations of relation paths. A path-constraint resource allocation algorithm is employed to measure the reliability of path. In this paper, we learn the semantic composition of dependency paths over dependency trees.

3 Method

In this section, we first present the unsupervised learning of word and dependency path embeddings. In addition, we describe how to enhance word embeddings by leveraging the linear context in a multi-task learning manner. Then, the construction of embedding features for CRF model is discussed.

3.1 Unsupervised Learning of Word and Dependency Path Embeddings

We learn the distributed representations of words and dependency paths in a unified model, as shown in Figure 1. We extract the triple \((w_1, w_2, r)\) from dependency trees, where \(w_1\) and \(w_2\) denote two words, the corresponding dependency path \(r\) is the shortest path from \(w_1\) to \(w_2\) and consists of a sequence of grammatical relations. For example, in Figure 1, given that service is \(w_1\) and professional is \(w_2\), we obtain the dependency path \(\ast \rightarrow \top \ast \rightarrow \top \ast \rightarrow \top \ast \rightarrow \ast\). We notice that considering the lexicalized dependency paths can provide more information for the embedding learning. However, we need to memorize more dependency path frequencies for the learning method (negative sampling) described in Section 3.3. Formally, the number of dependency paths is \(|V_{word}|^{n-1}|V_{dep}|^n\) when we take n-hop dependency paths into consideration.\(^1\)

It is computationally expensive, even when \(n = 2\). Hence, in our paper, we only include the grammatical relations for modeling \(r\) which also shows the promising results in the experiment section.

The proposed learning model connects two words with the dependency path in the vector space. By incorporating the dependency path into the model, the word embeddings encode more syntactic information [Levy and Goldberg, 2014] and the distributed representations of dependency paths are explicitly learned. To be specific, this model requires that \(r\) is close to \(w_2 - w_1\) and we minimize the following loss function for the model learning:

\[
\sum_{(w_1, w_2, r) \in C_1} \sum_{r \sim p(r)} \max \{0, 1 - (\mathbf{w}_2 - \mathbf{w}_1)^\top r + (\mathbf{w}_2 - \mathbf{w}_1)^\top r'\} \quad (1)
\]

where \(C_1\) represents the set of triples extracted from the dependency trees parsed from the text corpus, \(r\) is a sequence of grammatical relations that is denoted as \((g_1, g_2, \ldots, g_n)\), \(n\) is the hop number of \(r\), \(g_i\) is the \(i\)-th grammatical relation in \(r\), and \(p(r)\) is the marginal distribution for \(r\). The given loss function encourages that the triple \((w_1, w_2, r)\) has a higher ranking score than the randomly chosen triple \((w_1, w_2, r')\). The ranking score is measured by the inner product of \(r/r'\) and vector \(\mathbf{w}_2 - \mathbf{w}_1\).

\(^1\)\(V_{word}\) is the set of words and \(|V_{word}|\) is over 100,000 in our paper. \(V_{dep}\) is the set of grammatical relations and \(|V_{dep}|\) is about 50 in our paper.
We employ the recurrent neural network to learn the compositional representations for multi-hop dependency paths. The composition operation is realized with a matrix $W$:

$$ h_i = f(W[h_{i-1}; g_i]) $$

$$ f(x) = \begin{cases} 
1 & \text{if } x > 1 \\
 x & \text{if } -1 \leq x \leq 1 \\
-1 & \text{if } x < -1 
\end{cases} \quad (2) $$

where $f$ is a hard hyperbolic tangent function ($hTanh$), $[a; b]$ is the concatenation of two vectors, $g_i$ is the embedding of $g_i$. We set $h_1 = g_1$ and recursively perform composition operation to obtain the final representation $r = h_n$. Figure 1 (Left) illustrates the composition progress of the dependency path from service to professional. In this work, we train the triples whose hop numbers are less than or equal to 3. It is time-consuming to learn embeddings when we consider the triples with larger hop dependency paths.

### 3.2 Multi-Task Learning with Linear Context

We use the linear context to enhance word embeddings. The proposed method is based on the distributional hypothesis that words in similar contexts have similar meanings. Inspired by Skip-gram [Mikolov et al., 2013a], we enhance word embeddings by maximizing the prediction accuracy of the context word $c$ that occurs in the linear context of a target word $w$. The method is shown in Figure 1 (Right). Following the previous setting in [Mikolov et al., 2013b], we set the window size to 5. A hinge loss function, which has been proven to be effective in the cross-domain representation learning [Bollegala et al., 2015], is modified to the following function:

$$ \sum_{(c,w) \in C_2} \sum_{c' \sim p(w)} \max\{0, 1 - w^\top c + w^\top c'\} \quad (4) $$

where, $(c, w) \in C_2$ means that the word $c$ appears in the linear context of the target word $w$, and $p(w)$ is the marginal distribution for $w$. Every word has two roles, the target word and the context word of other target words. In this paper, we use different vectors to represent the two roles of a word.

### 3.3 Model Training

The negative sampling method [Collobert et al., 2011; Mikolov et al., 2013b] is employed to train the embedding model. Specifically, for each pair $(w, c)$, we randomly select $k_W$ words $c'$ that do not appear in the linear context window of the target word $w$. These randomly chosen words are sampled based on the marginal distribution $p(w)$ and the $p(w)$ is estimated from the word frequency raised to the $\frac{3}{4}$ power [Mikolov et al., 2013a] in the corpus. We use the same method to optimize the Equation (1). The difference is, we sample path $r'$ separately according to the hop number and randomly select $k_{R_2}$ dependency paths for the $n$-hop path $r$. The selected dependency paths $r'$ have the same hop number with $r$. We experimentally find that $k_W = 5$ is a trade-off between the training time and performance. Similarly, $k_{R_1}$, $k_{R_2}$ and $k_{R_3}$ are set to 5, 3 and 2.

The model is trained by back-propagation. We use asynchronous gradient descent for parallel training. Following the strategy for updating learning rate [Mikolov et al., 2013a], we linearly decrease it over our training instances. The initial learning rate is set to 0.001.

### 3.4 Aspect Term Extraction with Embeddings

We use the CRF model for aspect term extraction, and the features are constructed based on the learned embeddings. Besides the target word embedding features, we also utilize the features encoding the context information, (1) the linear context is within a fixed window of the target word in the linear sequence; (2) the dependency context is directly related with the target word in the dependency tree. These context features are used as complementary information to help label aspect terms.

Given a sentence $s = (w_1, w_2, ..., w_{|s|})$ and its dependency tree, we denote $C(w_i)$ as the function extracting the features for word $w_i$. In our case, $C(w_i) = [\text{Emb}(w_i), C_w(w_i), C_d(w_i)]$, where $\text{Emb}(w_i)$ represents the embedding of word $w_i$, $C_w(w_i)$ and $C_d(w_i)$ refer to the linear context embedding and the dependency context embedding respectively. These extracted embeddings are concatenated as CRF features.

Based on the assumption that the label of a word mainly depends on itself and its neighboring words [Collobert et al.,
Table 1: Statistics of the SemEval 2014 and 2015 datasets.

| Dataset       | # Sentences | # Aspects |
|---------------|-------------|-----------|
| D1-Train      | 3045        | 2358      |
| D1-Test       | 800         | 654       |
| D2-Train      | 3041        | 3693      |
| D2-Test       | 800         | 1134      |
| D3-Train      | 1315        | 1192      |
| D3-Test       | 685         | 678       |

Table 2: Statistics of unlabeled data.

| Dataset | # Sentences | # Tokens |
|---------|-------------|---------|
| Yelp    | 14M         | 310M    |
| Amazon  | 13M         | 280M    |

in-domain corpora for restaurant domain and laptop domain respectively. As the size of laptop reviews is small in Amazon corpus, we add the reviews of other similar products in laptop corpus. These products consist of Electronics, Kindle, Phone and Video Game. All corpora are parsed by using the Stanford corenlp⁶. The derived tokens are converted to lowercase, the words and dependency paths that appear less than 10 are removed. The details of unlabeled data are shown in Table 2.

In order to choose \( l \) and \( d \), we use 80% sentences in training data as training set, and the rest 20% as development set. The dimensions of word and dependency path embeddings are set to 100. Larger dimensions get similar results in the development set but need more training time. \( l \) is set to 15 that performs best in the development set.

We use an available CRF tool⁷ for tagging aspect terms, and set the parameters with default values. The only exception is the regularization option. L1 regularization is chosen as it obtains better results than L2 regularization in the development set.

4.2 Result and Analysis

We compare our method with the following methods:

(1) Naïve: Token in test sentences is tagged as an aspect term, if it is in the dictionary that contains all the aspect terms of training sentences.

(2) Baseline Feature Templates: Most of features used in the Baseline Feature Templates are adopted from the NER Feature Templates [Guo et al., 2014] which are described in Table 3.

(3) IHS_RD: IHS_RD is the top system in D1 which also relies on CRF with a rich set of lexical, syntactic and statistical features. Unlike other systems in SemEval 2014 and our work, IHS_RD trains the CRF model on the reviews of both the restaurant domain and laptop domain.

(4) DLIREC: DLIREC is the top system in D2 which relies on CRF with a variety of lexicon, syntactic, semantic features derived from NLP resources. In terms of features based on the dependency tree, they use the head word, head word POS and one-hop dependency relation of the target word. Besides, different from our work, DLIREC utilizes the corpora of Yelp and Amazon reviews to derive cluster features for words.

(5) EliXa: EliXa is the top system in D3 which addresses the problem using an averaged perceptron with a BIO tagging scheme. The features used in EliXa consist of n-grams, token shape (digits, lowercase, punctuation, etc.), previous prediction, n-gram prefixes and suffixes, and word clusters derived from additional data (Yelp for Brown and Clark clusters; Wikipedia for word2vec clusters).

⁶https://nlp.stanford.edu/software/corenlp.shtml
⁷https://crfsharp.codeplex.com/
We conduct experiments to evaluate our method of word and dependency path embeddings (WDEmb) with state-of-the-art embedding methods.

The baseline embedding algorithms include Dependency Recurrent Neural Language Model (DRNLM) [Piotr and Andreas, 2015], Skip-gram, CBOW [Mikolov et al., 2013a] and Dependency-based word embeddings (DepEmb) [Levy and Goldberg, 2014]. The DRNLM predicts the current words given the previous words, aiming at learning a probability distribution over sequences of words. Note that, besides the previous words, DRNLM also uses one-hop dependency relation as the additional features to help the model learning. Skip-gram learns word embeddings by predicting context word given the target word, while CBOW learns word embeddings by predicting current word given the context words. DepEmb learns word embeddings using one-hop dependency context. It is similar to our method to encode functional information into word embeddings. But the dependency path considered is one-hop rather than multi-hop.

The implementations of baseline models are publicly released\(^7\), and are used to train embeddings on the same corpora as our model. We set the parameters of baselines to ensure a fair comparison.\(^8\). As the baselines do not learn the distributed representation of multi-hop dependency path, we compare our model with them in the feature settings (W and W+L) where only word embeddings are used. The statistical significance tests are calculated by approximate randomization and the results are shown in Table 5.

Among these baselines, the DepEmb learns word embeddings by dependency context, and performs best in both domains. It indicates that the syntactic information is more desirable in aspect term extraction. WDEmb outperforms DepEmb and the reasons are that, (1) compared with DepEmb that embeds a unit word + dependency path into a vector, WDEmb learns the word and dependency path embeddings separately, and alleviates the data sparsity problem; (2) multi-hop dependency paths are taken into consideration in WDEmb which help encode more syntactic information. DRNLM performs worst among these baselines, though it utilizes the dependency relation in their model. The reason is that the DRNLM focuses on language modeling and does not include subsequent words which are used in other embedding baselines, as the context.

We also employ two methods for qualitative analysis of the learned embeddings, (1) we select the 6 most similar words (by cosine similarity) for each aspect word. Results are shown in Table 6; (2) we design some queries made up of one word and one dependency path, such as delicious +

\(^7\)DepEmb (https://levyomer.wordpress.com/software/)

\(^8\)The dimensions and interval numbers of baselines are tuned in the development set. The interval numbers of DRNLM, Skip-gram and CBOW are set to 10, and the interval number of DepEmb is set to 15. All dimensions of baselines are set to 100. The window sizes of Skip-gram and CBOW are set to 5, which is the same as our model of multi-task learning with linear context.
Table 5: Results of different embedding methods. The marker * represents p-value < 0.05, the marker † represents p-value < 0.01, and the compared baseline is DepEmb.

Table 6: Aspect words and their most similar words. The words that are less similar with the aspect words in syntactic function are highlighted in red. The words that are less similar with the aspect words in topical information are highlighted in blue.

Table 7: Queries and their most similar words.

5 Conclusion and Future Work

In this paper, we propose to learn the distributed representations of words and dependency path in an unsupervised way for aspect term extraction. Our method leverages the dependency path information to connect words in the embedding space. It is effective to distinguish words with similar context but different syntactic functions which are important for aspect term extraction. Furthermore, the distributed representation of multi-hop dependency paths are also derived explicitly. The embeddings are then discretized and used as features in the CRF model to extract aspect terms from review sentences. With only the distributed representation features, we obtain comparable results with the top systems in aspect term extraction on the benchmark dataset. Recently, [Wang et al., 2015a] propose to incorporate knowledge graph to represent the documents or short text [Wang et al., 2015b; 2016]. This could be interesting to combine this representation with our embedding methods to further improve the performance.

6 Acknowledgments

This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Numbers 61272343, 61472006), the Doctoral Program of Higher Education of China (Grant No. 20130001110032) as well as the National Basic Research Program (973 Program No. 2014CB340405),
References

[Bengio et al., 2013] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *PAMI*, 35(8):1798–1828, Aug 2013.

[Bollegala et al., 2015] Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. Unsupervised cross-domain word representation learning. *arXiv:1505.07184*, 2015.

[Bordes et al., 2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344, 2011.

[Chernyshevich, 2014] Maryna Chernyshevich. Ihs r&d belar: Cross-domain extraction of product features using conditional random fields. *SemEval*, page 309, 2014.

[Collobert et al., 2011] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *JMLR*, 2011.

[Guo et al., 2014] Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. Revisiting embedding features for simple semi-supervised learning. In *EMNLP*, pages 110–120, 2014.

[Hu and Liu, 2004] Mingqing Hu and Bing Liu. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760, 2004.

[Jakob and Gurevych, 2010] Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *EMNLP*, pages 1035–1045, 2010.

[Levy and Goldberg, 2014] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *ACL*, pages 302–308, 2014.

[Li et al., 2010] Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *ACL*, pages 653–661, 2010.

[Li et al., 2012] Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. Cross-domain co-extraction of sentiment and topic lexicons. In *ACL*, pages 410–419, 2012.

[Lin et al., 2015] Yankai Lin, Zhiyuan Liu, Huainbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*, 2015.

[Liu et al., 2015a] Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, pages 1433–1443, 2015.

[Liu et al., 2015b] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. A dependency-based neural network for relation classification. In *ACL*, pages 285–290, 2015.

[Mikolov et al., 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010.

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

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

[Neelakantan et al., 2015] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. In *ACL*, pages 156–166, July 2015.

[Piotr and Andreas, 2015] Mirowski Piotr and Vlachos Andreas. Dependency recurrent neural language models for sentence completion. *arXiv:1507.01193*, 2015.

[Pontiki et al., 2014] Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval*, pages 27–35, 2014.

[Pontiki et al., 2015] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval*, 2015.

[Qiu et al., 2011] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *CL*, 37(1):9–27, 2011.

[San Vicente et al., 2015] Ifaki San Vicente, Xabier Saralegi, and Rodrigo Agerri. Elixa: A modular and flexible absa platform. In *SemEval*, pages 748–752, 2015.

[Toh and Wang, 2014] Zhiqiang Toh and Wenting Wang. Dlirec: Aspect term extraction and term polarity classification system. 2014.

[Turian et al., 2010] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *ACL*, pages 384–394, 2010.

[Wang et al., 2015a] Chenguang Wang, Yangqiu Song, Ahmed El-Kishky, Dan Roth, Ming Zhang, and Jiawei Han. Incorporating world knowledge to document clustering via heterogeneous information networks. In *KDD*, pages 1215–1224, 2015.

[Wang et al., 2015b] Chenguang Wang, Yangqiu Song, Horan Li, Ming Zhang, and Jiawei Han. Knowsim: A document similarity measure on structured heterogeneous information networks. In *ICDM*, pages 1015–1020, 2015.

[Wang et al., 2016] Chenguang Wang, Yangqiu Song, Horan Li, Ming Zhang, and Jiawei Han. Text classification with heterogeneous information network kernels. In *AAAI*, 2016.

[Yeh, 2000] Alexander Yeh. More accurate tests for the statistical significance of result differences. *COLING*, pages 947–953, 2000.