

Learning Latent Sentiment Scopes for Entity-Level Sentiment Analysis

Hao Li, Wei Lu

Singapore University of Technology and Design
 8 Somapah Road, Singapore, 487372
 hao_li@mymail.sutd.edu.sg, luwei@sutd.edu.sg

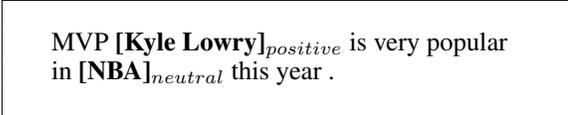
Abstract

In this paper, we focus on the task of extracting named entities together with their associated sentiment information in a *joint* manner. Our key observation in such an *entity-level sentiment analysis* (a.k.a. *targeted sentiment analysis*) task is that there exists a *sentiment scope* within which each named entity is embedded, which largely decides the sentiment information associated with the entity. However, such sentiment scopes are typically not explicitly annotated in the data, and their lengths can be unbounded. Motivated by this, unlike traditional approaches that cast this problem as a simple sequence labeling task, we propose a novel approach that can explicitly model the latent sentiment scopes. Our experiments on the standard datasets demonstrate that our approach is able to achieve better results compared to existing approaches based on conventional conditional random fields (CRFs) and a more recent work based on neural networks.

Introduction

Designing effective algorithms that can automatically perform *sentiment analysis*, the task of inferring the underlying sentiment-level information associated with texts, is an important yet challenging task that has applications in various fields (Pang and Lee 2008; Liu 2010; Ortigosa, Martín, and Carro 2014; Smailović et al. 2013; Li and Wu 2010). Traditional sentiment analysis algorithms largely focus on assigning sentiment information at the complete sentence and document level. Recently, there is a growing interest in performing sentiment analysis at the target entities, where two different types of tasks are considered: 1) to determine sentiment information for entities which are already recognized, 2) to jointly decide the named entities and their respective sentiment information. In this paper, we focus on the latter, which is a more challenging task than the former. Such a task is also called *entity-level sentiment analysis* or *targeted sentiment analysis* (Mitchell et al. 2013), since the sentiment information is assigned to specific target entities – certain named entities in the texts. Figure 1 presents a real example, where entities of interest are highlighted with their sentiment information. Such an entity-level sentiment analysis task is more challenging, but is more useful in many applications such as product review (Hu and Liu 2004).

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



MVP [Kyle Lowry]_{positive} is very popular in [NBA]_{neutral} this year .

Figure 1: Entity-level sentiment analysis with named entities and sentiment highlighted.

Currently, existing models tackled the entity-level sentiment analysis problem from a sequence labeling perspective, attempting to assign each word token in the text a specific tag. Such tags were used to indicate boundary and sentiment information associated with entities in the text. The sequence labeling models can be either based on the linear-chain or the factorial conditional random fields (CRFs) (Mitchell et al. 2013), or certain neural structured prediction models (Zhang, Zhang, and Vo 2015). Such models can typically make use of features based on information surrounding each word token within a fixed window of bounded size. Thus the basic assumption behind such approaches is the sentiment of each entity can be fully determined based on such local contextual information.

However, such an assumption is not always true in practice. First, the right information required for determining the current entity’s sentiment can appear outside the window. Second, the information appearing inside the window is not always indicative of the true sentiment associated with the entity. Consider the example shown in Figure 1 with two entities **Kyle Lowry** and **NBA**. Assume we use a bounded window of size 2 (*i.e.* when constructing features, previous and next 2 word tokens are considered), the word *popular* that conveys key sentiment information will appear outside of the window used for predicting the sentiment of the entity **Kyle Lowry**. Thus such a model would have difficulty assigning the positive sentiment to this entity. Increasing the window size will not completely resolve this issue as the sentiment-indicative keywords can appear far away from the entity (*e.g.*, consider the sentence “*David Copperfield is believed to be one of the most amazing magicians ...*”. Here the keyword “*amazing*” appears far away from the entity “*David Copperfield*”). On the other hand, since the same word *popular* appears inside the window of the next entity, it would be used to wrongly predict the sentiment associ-

ated with the entity **NBA**, which in fact should be assigned a neutral sentiment in this example.

Based on the above observations, in this work we propose a novel approach to entity-level sentiment analysis, by assuming there exists a variable-size window of words surrounding each entity that determine its sentiment polarity, where the size of such windows can be unbounded. Such window information is typically not available and needs to be learned from data. We design a model that can effectively capture entity, targeted sentiment, as well as such window information in a joint manner. Specifically, we make the following major contributions in this work:

- We introduce the novel notion of *sentiment scopes* for the task of entity-level sentiment analysis. Based on this, we propose novel models that are able to efficiently learn sentiment scopes from data to jointly predict named entities and their associated sentiment information.
- Empirically, we demonstrate that the proposed model is able to consistently outperform previous approaches based on conventional models based on CRF and neural networks.

Approach

We present our approach to entity-level sentiment analysis in this section. We first formally introduce the notion of sentiment scopes. Next we discuss our models.

Sentiment Scopes

Following previous work (Mitchell et al. 2013), only volitional entities (named entities of type *Person* and *Organization*) are considered for sentiment analysis in this work. Our primary assumption is that for each volitional entity, there exists a *sentiment scope*, which is analogous to the notion of *negation scope* (Zou, Zhou, and Zhu 2013). The sentiment scope fully determines the entity’s sentiment-level information. We define a sentiment scope of a particular entity e in a given text as a consecutive sequence of words, which contains the entity e , as well as words surrounding e . Unlike existing approaches which simply assume there exists a window of fixed size around the volitional entity, we assume there exists a window of unbounded size that specifies the boundaries of each sentiment scope. Note that we also assume each word appearing in a given sentence strictly belongs to exactly one sentiment scope of a particular volitional entity. This means the sentiment scopes cover the complete sentence, while they do not overlap with one another.

However, the sentiment scopes are not explicitly annotated in the training data. We thus need to build models that can automatically learn such latent information from the data. As we will see later, with the above assumptions on the properties of sentiment scopes in each sentence, we will be able to design compact representations encompassing exponentially many possible combinations of sentiment scopes for efficient inference.

Let us return to the example shown in Figure 1 in the previous section. The sentence, which is a tweet, contains two volitional entities: **Kyle Lowry** and **NBA**, where the former

should be associated with a positive sentiment and the latter with a neutral one. In this case, the keywords *very* and *popular* that are indicative of a positive sentiment should be used to predict the sentiment for the entity **Kyle Lowry** rather than **NBA**, even though the word *popular* appears closer to the latter. Thus, the correct latent sentiment scope for **Kyle Lowry** in this case could be *MVP Kyle Lowry is very popular*, while the sentiment scope for **NBA** could be *in NBA this year*. We will see how our model can capture such latent information.

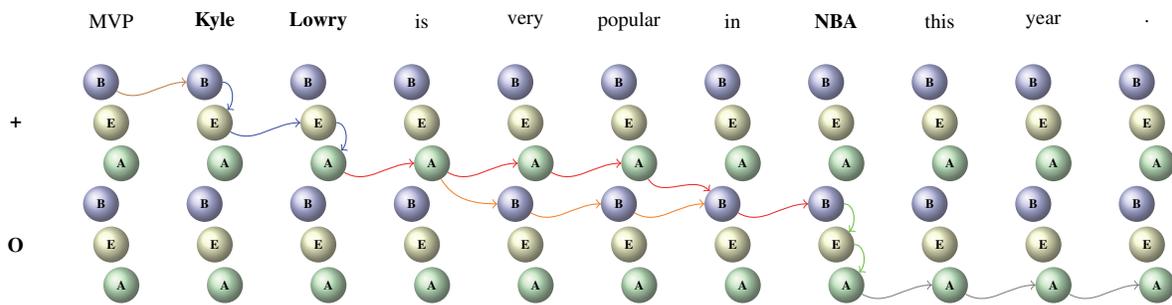
Models

We adopt an approach based on graphical models together with the above assumptions on sentiment scopes. Similar to the collapsed CRF, we integrate both named entity information and sentiment level information together to form label sequences. We extend such an approach by using 9 different types of nodes at each word/position, namely \mathbf{B}_+ , \mathbf{E}_+ and \mathbf{A}_+ nodes for positive sentiment, \mathbf{B}_- , \mathbf{E}_- and \mathbf{A}_- nodes for negative sentiment, as well as \mathbf{B}_o , \mathbf{E}_o and \mathbf{A}_o nodes for neutral sentiment.

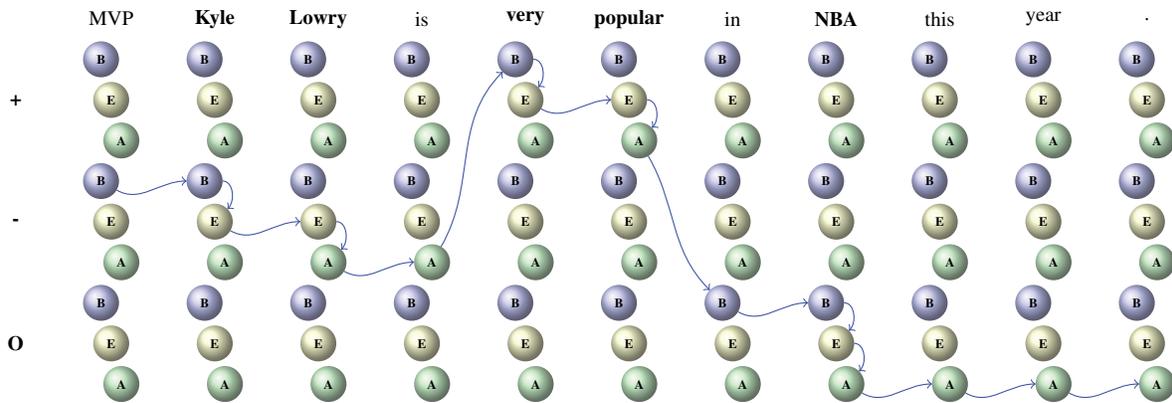
The \mathbf{B} nodes are used to denote that the current word is part of a sentiment scope of a certain sentiment polarity, but it appears before the named entity in the scope or exactly as the first word in the entity. The \mathbf{E} nodes are used to denote the fact that the current word is part of a named entity of a certain polarity. The \mathbf{A} nodes are used to denote that the current word appears within a sentiment scope of a certain polarity but is after the named entity or as the last word in the named entity. At each position, only some relevant nodes will be selected based on where the named entities and sentiment scopes are. The nodes selected at different positions will then be connected with each other, forming a *sentiment scope graph* that shows the named entity information and their associated sentiment scopes with polarity information.

We show illustrative examples in Figure 2. At each position, there are 9 different nodes that can be divided into 3 groups. Each group represents one possible sentiment: positive (+), negative (-), and neutral (o). We again use the same sentence “*MVP Kyle Lowry is very popular in NBA this year*.” as an example. This example contains two entities. Thus two sentiment scopes should be formed for this sentence, with the first having the positive sentiment and the second a neutral one.

Let us now take a closer look at how exactly a particular sentiment scope graph can be formed. In this example, the first word is “*MVP*” that is not part of any entity. However this word appears within the sentiment scope of an entity with the positive sentiment. Thus the node \mathbf{B} with a *positive* sentiment is selected. The next word appears as the first word of the named entity “*Kyle Lowry*”, and therefore the node \mathbf{B} is selected. Also, since this word “*Kyle*” is part of a named entity, the node \mathbf{E} is also selected at this position. The next word “*Lowry*” is again part of a named entity, but it appears as the last word in the entity. Therefore both \mathbf{E} and \mathbf{A} nodes of the positive sentiment are selected. The next word is “*is*”, which appears outside of any entity. If we assume this word is still part of the sentiment scope for the



a. A sentiment scope graph consisting of correct named entities and sentiment annotation with two possible latent sentiment scopes.



b. An example sentiment scope graph associated with incorrect named entities and incorrect sentiment annotation.

Figure 2: Example sentiment scope graphs.

named entity “Kyle Lowry”, the node **A** at the corresponding position needs to be selected, as shown in Figure 2 (a). The next word is “very”. We can still assume this word appears within the sentiment scope of the first entity, but we can also assume it appears within the sentiment scope of the second entity (“NBA”) with a neutral polarity. In the former case, the node **A** needs to be selected, while in the latter case the node **B** of the neutral sentiment will be selected.

We can continue the above node-selection process until we have reached the end of the sentence. Next we can connect the selected nodes at different positions to form a sentiment scope graph, which is a directed chain. In Figure 2 (a), two possible graphs among all the possible graphs are highlighted for the above example sentence.

Note that given any sentence, there exist exponentially many possible sentiment scope graphs, each specifying a different possible entity and sentiment scope combination. For example, in Figure 2 (b), we show another example sentiment scope graph that comes with incorrect sentiment scope information. Specifically, in this graph, the scope “very popular” is wrongly annotated as the second entity with the positive sentiment.

Our task is to build a model that can assign high scores to those sentiment scope graphs with correct named entity and sentiment scope information, and can assign lower scores to other incorrect sentiment scope graphs. Following the CRF

model (Lafferty, McCallum, and Pereira 2001), we use a log-linear formulation with latent variables to parameterize our model. Specifically, the probability of predicting a possible output y , which is a collection of named entities and their associated sentiment information, given an input sentence x is defined as:

$$p(y|x) = \frac{\sum_{\mathbf{h}} \exp(\mathbf{w}^T \mathbf{f}(x, y, \mathbf{h}))}{\sum_{y', \mathbf{h}'} \exp(\mathbf{w}^T \mathbf{f}(x, y', \mathbf{h}'))} \quad (1)$$

where \mathbf{w} is the weight vector, and $\mathbf{f}(x, y, \mathbf{h})$ is the feature vector defined over the sentence x and the output structure y , together with the latent variable \mathbf{h} that provide the detailed sentiment scope information for the (x, y) tuple.

A Semi-Markov Variant

The above model relies on the first-order Markov assumptions between nodes of adjacent words. We can relax this assumption, resulting in a semi-Markov (Sarawagi and Cohen 2004) variant of the above model. Different from the original model, in this new model we do not have the **E** nodes. Whenever we have an entity, we establish a direct edge that connects the first word’s **B** node and the last word’s **A** node. An illustrative example is shown in Figure 3. In this example, the words “Internet Marketing Agency” is an entity. Thus an edge between the **B** node of “Internet” and **A** node of “Agency” is established to reflect this fact. The rest of the

sentiment scope graph remains the same. Such a model is able to capture certain non-Markovian features that can not be captured by the standard model above. However, it comes with a slightly higher time complexity.

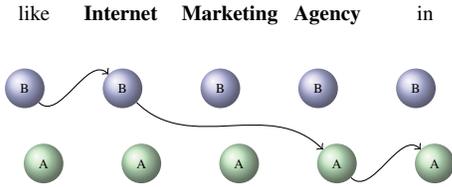


Figure 3: An example partial sentiment scope graph for the semi-Markov variant, where **Internet Marketing Agency** is an entity.

Features

Feature functions can be factorized as products of two indicator functions: one defined on the input sequence (input features) and the other on the output labels (output features). In other words, we could write each feature $f_j(\mathbf{x}, \mathbf{y}, \mathbf{h})$ as $f_k^{in}(\mathbf{x}) \times f_l^{out}(\mathbf{y}, \mathbf{h})$. Following Mitchell et al. (2013) and Zhang et al. (2015), we created all input features based on Table 1, and used the MPQA lexicon (Wilson, Wiebe, and Hoffmann 2005) and the SentiWordNet lexicon (Baccianella, Esuli, and Sebastiani 2010) to obtain polarity information for English. The Spanish versions of the MPQA lexicon and SentiWordNet lexicon are created based on their English versions. These lexicons are exactly the same as those used in (Mitchell et al. 2013). Features for the semi-Markov variant are listed in the supplementary material.

Our models are able to encode features defined over latent sentiment scopes. Specifically, we define features as combinations of each non-entity word and the sentiment information of the sentiment scope it belongs to. We also create features as combinations of each non-entity word, its sentiment information, as well as its relative position to the named entity (*i.e.*, *before* or *after* the named entity).

Training and Decoding

We aim to minimize the negative joint log-likelihood of our dataset with L_2 regularization, which is defined as:

$$\mathcal{L}(\mathbf{w}) = \sum_i \log \sum_{\mathbf{y}', \mathbf{h}'} \exp(\mathbf{w}^T \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}', \mathbf{h}')) - \sum_i \log \sum_{\mathbf{h}} \exp(\mathbf{w}^T \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{h})) + \lambda \mathbf{w}^T \mathbf{w} \quad (2)$$

where $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ is the i -th training instance and λ is the L_2 regularization parameter. Here the output \mathbf{y} basically conveys the information annotated in the dataset, including entity boundaries and sentiment information, but does not provide the latent sentiment scope information. Due to the latent variables involved, the above objective function is non-convex. The gradient with respect to each parameter w_k can

Surface Features
binned word length, message length, sentence position; Jerboa features; word identity; word lengthening; punctuation characters, has digit; has dash; is lower case; is 3 or 4 letters; first letter capitalized; more than one letter capitalized, etc.
Linguistic Features
function words; can syllabify; curse words; laugh words; words for good/bad; slang words; abbreviations; intensifiers; subjective suffixes and prefixes (such as diminutive forms); common verb endings; common noun endings
Brown Clustering Features
cluster at length 3; cluster at length 5
Sentiment Lexicon Features
is sentiment-bearing word; prior sentiment polarity

Table 1: Features used in our models. All these features are exactly the same as those discrete features used in Mitchell et al. (2013) and Zhang et al. (2015).

be computed as:

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_k} = \sum_i \mathbf{E}_{p(\mathbf{y}', \mathbf{h}' | \mathbf{x}^{(i)})} \left[f_k(\mathbf{x}^{(i)}, \mathbf{y}', \mathbf{h}') \right] - \sum_i \mathbf{E}_{p(\mathbf{h} | \mathbf{x}^{(i)}, \mathbf{y}^{(i)})} \left[f_k(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{h}) \right] + 2\lambda w_k \quad (3)$$

where $\mathbf{E}_p[\cdot]$ is the expectation under the distribution p .

We can use standard gradient-based methods to optimize the objective function. In this work, we choose to use L-BFGS (Liu and Nocedal 1989) as the optimization algorithm, which was previously shown effective in optimizing similar objective functions (Blunsom, Cohn, and Osborne 2008).

To efficiently calculate the objective value and the above expectations, we apply a generalized forward-backward style algorithm, which allows us to perform exact inference using dynamic programming. The key observation here is that there exists a topological ordering amongst all the nodes appearing in any sentiment scope graph. This allows us to use a compact representation to encode exponentially many sentiment scope graphs for a given sentence, and the resulting representation is a directed acyclic graph.

During decoding, we use standard MPE inference, where we simply replace the sum operation in the forward-backward style algorithm used during training by the max operation. The resulting algorithm is analogous to the Viterbi algorithm used for decoding a linear-chain CRF. From the decoded sentiment scope graph, we can simply read off the predicted named entities and their associated sentiment information. Our model is also able to obtain the predicted sentiment scope information associated with each predicted named entity as a by-product.

The time complexity of the inference procedure for the sentiment scope model is $O(nT^2)$, where n is the sentence length, and T is the number of different sentiment types, which in our case is a small constant (3). For the semi-Markov model, the time complexity is $O(nLT^2)$, where L is the maximal entity length.

Model	English						Spanish					
	Entity Recognition			Sentiment Analysis			Entity Recognition			Sentiment Analysis		
	<i>P.</i>	<i>R.</i>	<i>F</i> ₁									
Pipeline	65.74	47.59	55.18	46.80	33.87	39.27	71.29	58.26	64.11	43.80	35.80	39.40
Collapsed	54.00	42.69	47.66	38.40	30.38	33.90	62.20	52.08	56.66	39.39	32.96	35.87
Joint	59.45	43.78	50.32	41.77	30.80	35.38	66.05	52.55	58.51	41.54	33.05	36.79
(Zhang, et al. 2015)	60.69	51.63	55.67	43.71	37.12	40.06	70.23	62.00	65.76	45.99	40.57	43.04
SS	63.18	51.67	56.83	44.57	36.48	40.11	71.49	61.92	66.36	46.06	39.89	42.75
SS (+word embeddings)	66.35	56.59	61.08	47.30	40.36	43.55	73.13	64.34	68.45	47.14	41.48	44.13
SS (+POS tags)	65.14	55.32	59.83	45.96	39.04	42.21	71.55	62.72	66.84	45.92	40.25	42.89
SS (semi)	63.93	54.53	58.85	44.49	37.93	40.94	70.17	64.15	67.02	44.12	40.34	42.14

Table 2: Main results. (SS: sentiment scope model introduced in this paper, semi: the semi-Markov variant of the model)

Experimental Setup

Data

We mainly performed most of our experiments based on the dataset from (Mitchell et al. 2013). This dataset consists of 7,105 Spanish tweets and 2,350 English tweets, with named entities and their sentiment information annotated. See supplementary material for detailed corpus statistics. Following previous research efforts (Mitchell et al. 2013; Zhang, Zhang, and Vo 2015), we report 10-fold cross-validation results, and split 10% of the training set for development.

We also carried out additional experiments based on datasets from SemEval 2016 and TASS 2015. Such results are listed in the supplementary material.

Evaluation Metrics

Different from (Mitchell et al. 2013) which performed evaluation at the word token level, we conducted evaluations at the entity level. This makes our results comparable with those of (Zhang, Zhang, and Vo 2015), which also adopted the same evaluation methodology. Specifically, for entity recognition, we calculated *precision* (*P.*, the percentage of entities predicted by the model which are correct), *recall* (*R.*, the percentage of the entities in the dataset that are correctly discovered by the model) and *F*₁-*measure* (the harmonic mean of precision and recall). We also report results for entity-level sentiment analysis (where the prediction is regarded as correct only if both the entity and sentiment information are correct), and subjectivity.

Baseline

We first compared against three CRF-based baselines: the collapsed model, the joint model and the pipeline model, which are also the models used in (Mitchell et al. 2013). We re-implemented all these models, and used exactly the same word-level features in all models to have a fair comparison.

The collapsed model is essentially a linear-chain CRF model, where the tagset **{O, B-positive, B-negative, B-neutral, I-positive, I-negative, I-neutral}** was used. Each tag indicates the sentiment information as well as the relative position of the word in the entity. The joint model is essentially a factorial CRF model (Sutton, McCallum, and Rohanimanesh 2007) with two layers, which jointly labels each sentence with two different tagsets: **{B, I, O}**, and **{positive,**

negative, neutral, no sentiment}. The pipeline model involves two linear-chain CRF. First, we use the tagset **{B, I, O}** to perform volitional entity recognition. Next, we predict the sentiment information for each named entity. Finally we combine results from these two steps to output the named entities with targeted sentiment.

Features used in the collapsed model, joint model and the first step of the pipeline model are the same as those described in Table 1. In the second step of the pipeline model, only sentiment related features are used to predict targeted sentiment information.

Another baseline we compared against is the best-performing model presented in (Zhang, Zhang, and Vo 2015), which is motivated by the neural CRF model (Do, Arti, and others 2010), integrating standard discrete CRF features and continuous features (based on word embeddings). Their “Pipeline Integrated” system that incorporates both discrete features and continuous features gives the best F-score in that paper. Note that their continuous features based on word embeddings are additional features, which are not available to us.

Results and Discussions

Comparisons with Baselines

Main results for both English and Spanish datasets are shown in Table 2. When comparing with the baseline pipeline, joint and collapsed models, our model consistently gives a significantly higher *F*₁-measure for both English and Spanish ($p < 10^{-4}$ under paired t-test for all cases except for entity recognition for both languages, in which case, comparing our model and pipeline, we have $p < 10^{-3}$). These results confirm that our model has the capability to capture some dependencies between entities and their sentiment information more accurately through the use of sentiment scopes (SS).

We also compared against the best performing system reported in (Zhang, Zhang, and Vo 2015). Note that such a comparison is unfair for us since in their model additional word embedding features (learned from their own Twitter data) were used, while we only used discrete features. Nevertheless, compared with this model, for both datasets, our model performs competitively with their best-performing “Pipeline Integrated” approach reported in (Zhang, Zhang, and Vo 2015).

Model	English						Spanish					
	Entity Recognition			Sentiment Analysis			Entity Recognition			Sentiment Analysis		
	<i>P.</i>	<i>R.</i>	<i>F</i> ₁									
SS	63.18	51.67	56.83	44.57	36.48	40.11	71.49	61.92	66.36	46.06	39.89	42.75
SS (no s.p.)	63.39	51.40	56.76	44.37	36.00	39.74	71.97	61.18	66.13	44.19	37.55	40.60
SS (fixed scopes)	62.11	49.93	55.34	43.69	35.15	38.94	69.70	58.64	63.69	44.84	37.73	40.97
SS (fixed scopes, no s.p.)	61.66	49.32	54.79	43.98	35.18	39.08	69.85	57.87	63.29	43.33	35.89	39.26
SS (semi)	63.93	54.53	58.85	44.49	37.93	40.94	70.17	64.15	67.02	44.12	40.34	42.14
SS (semi, fixed scopes)	65.49	54.06	59.21	45.11	37.24	40.79	69.64	62.69	65.97	42.94	38.67	40.69

Table 3: Results for additional experiments. (SS: our sentiment scope model introduced in this paper. semi: its semi-Markov variant. no s.p.: no sentiment propagation.)

To understand the effect of using word embeddings as features, we used the pre-trained English and Spanish word embeddings from polyglot (Al-Rfou, Perozzi, and Skiena 2013) which is publicly available, and constructed continuous features in our model (when constructing such features, the values at each dimension of the embeddings are regarded as feature values). Results show that inclusion of such features can consistently further improve our results, leading to the best results compared with all other systems. Experiments on the semi-Markov variant of our model show that such a model is effective as well, with comparable or slightly better results as compared to our standard model (we added additional non-Markovian features to indicate entity boundaries and relative positions for each word in the entity. We tuned L using the development set, where $L = 6$ for English, and $L = 7$ for Spanish).

Results also show that using POS tags as features can lead to improved results. We believe this is due to the fact that such POS tag information can be used to partially disambiguate the sentiment senses of certain words (e.g., the word “like” may have different sentiment senses when different POS tags are assigned to it).

We also measured the precision, recall and F-measure of subjectivity and non-neutral polarities as shown in Table 4 on the Spanish dataset and compared against the results reported in (Zhang, Zhang, and Vo 2015). The subjectivity measures whether a target sentence or phrase expresses an opinion or not according to (Liu 2010). Comparing with the best-performing system’s results reported in (Zhang, Zhang, and Vo 2015), both of our models can obtain a higher F-measure, especially when the non-neutral sentiment analysis is considered.

To understand the results qualitatively, we used our model to output the predicted sentiment scopes. One interesting example that we found was: “*When it comes to presentation, Mark Zuckerberg is no Steve Jobs ;P http://t.co/3ORETf2 facebook announcement*”. Here our model successfully marked “*Mark Zuckerberg*” as negative, and “*Steve Jobs*” as positive, due to the correct sentiment scope boundary predicted (between “*no*” and “*Steve*”).

Additional Experiments

To understand the effectiveness of our models better, we conducted some additional experiments on different variants of our sentiment scope model. Results are shown in Table 3.

Model	Subj (+,-,o)			SA (+,-)		
	<i>P.</i>	<i>R.</i>	<i>F</i> ₁	<i>P.</i>	<i>R.</i>	<i>F</i> ₁
(Zhang, et al.)	49.2	42.1	45.3	40.9	21.6	27.9
SS	49.0	42.4	45.4	37.2	24.7	29.6
SS (+emb)	50.0	44.0	46.8	37.6	25.4	30.2

Table 4: Results on subjectivity as well as non-neutral sentiment analysis on the Spanish dataset. (Subj(+,-,o): subjectivity for all polarities. SA(+,-): sentiment analysis for non-neutral polarities. SS: our basic sentiment scope model introduced in this paper without word embedding features. SS (+emb): SS model with word embedding features.

First, we removed all sentiment-level features defined at non-entity words in the sentiment scopes. This prevented the sentiment information associated with entities from being *propagated* to other words within the same sentiment scope. We call the resulting approach a model without “sentiment propagation” (*no s.p.*). Under such a setting, only sentiment features that appear around the entities within a fixed-size window can be exploited, while certain long-distance sentiment dependencies that can be captured by the sentiment scopes will not be captured. For both languages, the overall results of such an approach are much lower than those of the standard model. These results indicate that there does exist some long-distance sentiment information that needs to be captured, and can be captured by our model.

In another experiment, to understand the importance of modeling the sentiment scopes as latent variables, we implemented a simpler version of our model by assuming the boundaries of two adjacent sentiment scopes are fixed at the middle point of the sequence of words appearing in between two adjacent entities (*fixed scopes*). The results show such a model in general yields much worse results than the standard approach across two languages for both tasks. The only exception is when the semi-Markov model is considered and English data is used, where fixing the sentiment scopes can yield comparable results in F-measure. These results largely confirm the importance of modeling the latent sentiment scopes.

Related Work

Sentiment analysis and opinion mining has been a very popular task in the past decade (Pang and Lee 2008; Liu 2010). They can be largely regarded as structured prediction prob-

lems. In fact, there exist several approaches that rely on graphical models, especially sequence labeling models for such a task. For example, Jin et al. (2009) used a lexicalized HMM to extract opinions and their orientation. Li et al. (2010) used a CRF to perform joint extraction of opinions and opinion targets. Yang and Cardie (2012) proposed to use a semi-Markov CRF model for opinion expression extraction.

Mitchell et al. (2013) introduced the task of open domain targeted sentiment analysis, and regarded it as a sequence labeling problem. They proposed several models based on CRF that can predict both entity and targeted sentiment information. Zhang, Zhang and Vo (2015) proposed an approach that combines neural networks with a structured prediction model for open domain targeted sentiment task. They have demonstrated that it is useful to integrate both discrete and continuous features for such a task. The model learns to predict at each position the label based on contextual words from a window of limited size. The authors also proposed a recent model based on gated neural networks (Zhang, Zhang, and Vo 2016) to capture the influence of the surrounding words when performing sentiment classification of entities. While this approach shares similar motivations as ours, it tackles a simpler task where the named entities are assumed to be already recognized.

Other approaches to predicting sentiment structures also exist. For example, Socher et al. (2013) performed sentiment analysis on top of syntactic tree structures and introduced a sentiment treebank. They proposed to use neural networks to recursively predict sentiment labels at each tree node.

Conclusion

In this work, we explored entity-level sentiment analysis by using a novel model that captures latent sentiment scopes. The model yields significantly better results than those of different baseline systems, due to its ability to capture unbounded long-distance sentiment dependencies between words and entities. We also performed further investigations to verify the effectiveness of the proposed model. Future work includes extending the model to support other types of inputs such as trees (Socher et al. 2013), and to relax some assumptions we have made in this work so as to handle more sophisticated language phenomena. For example, one assumption that we made when defining the sentiment scopes was that they strictly do not overlap with one another. However, such an assumption is not always true in practice, and we believe some recent models developed for predicting overlapping structures (Lu and Roth 2015; Muis and Lu 2016) can be potentially used for modeling overlapping sentiment scopes.

We make our code, system and supplementary material available at <http://statnlp.org/research/st/>.

Acknowledgement

We would also like to thank the anonymous reviewers for their helpful comments. This work is supported by MOE Tier 1 grant SUTDT12015008.

References

- Al-Rfou, R.; Perozzi, B.; and Skiena, S. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proc. of CoNLL*.
- Baccianella, S.; Esuli, A.; and Sebastiani, F. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10.
- Blunsom, P.; Cohn, T.; and Osborne, M. 2008. A discriminative latent variable model for statistical machine translation. In *Proc. of ACL*.
- Do, T.; Arti, T.; et al. 2010. Neural conditional random fields. In *Proc. of AISTATS*.
- Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proc. of ACM SIGKDD*. ACM.
- Jin, W.; Ho, H. H.; and Srihari, R. K. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*. Citeseer.
- Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.
- Li, N., and Wu, D. D. 2010. Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision support systems* 48(2).
- Li, F.; Han, C.; Huang, M.; Zhu, X.; Xia, Y.-J.; Zhang, S.; and Yu, H. 2010. Structure-aware review mining and summarization. In *Proc. of COLING*.
- Liu, D. C., and Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming* 45(1-3).
- Liu, B. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing* 2.
- Lu, W., and Roth, D. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proc. of EMNLP*.
- Mitchell, M.; Aguilar, J.; Wilson, T.; and Van Durme, B. 2013. Open domain targeted sentiment. In *Proc. of EMNLP*.
- Muis, A. O., and Lu, W. 2016. Learning to recognize discontinuous entities. In *Proc. of EMNLP*, 75–84.
- Ortigosa, A.; Martín, J. M.; and Carro, R. M. 2014. Sentiment analysis in facebook and its application to e-learning. *Computers in Human Behavior* 31.
- Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2(1-2).
- Sarawagi, S., and Cohen, W. W. 2004. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*.
- Smailović, J.; Grčar, M.; Lavrač, N.; and Žnidaršič, M. 2013. Predictive sentiment analysis of tweets: A stock market application. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer.
- Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models

for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

Sutton, C.; McCallum, A.; and Rohanimanesh, K. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research* 8(Mar).

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of EMNLP*. Association for Computational Linguistics.

Yang, B., and Cardie, C. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proc. of EMNLP*.

Zhang, M.; Zhang, Y.; and Vo, D.-T. 2015. Neural networks for open domain targeted sentiment. In *Proc. of EMNLP*.

Zhang, M.; Zhang, Y.; and Vo, D.-T. 2016. Gated neural networks for targeted sentiment analysis. In *Proc. of AAAI*.

Zou, B.; Zhou, G.; and Zhu, Q. 2013. Tree kernel-based negation and speculation scope detection with structured syntactic parse features. In *Proc. of EMNLP*.