Neural Networks for Open Domain Targeted Sentiment

Meishan Zhang and **Yue Zhang** and **Duy-Tin Vo**
Singapore University of Technology and Design
{meishan_zhang, yue_zhang}@sutd.edu.sg,
duytin_vo@mymail.sutd.edu.sg

Abstract

Open domain targeted sentiment is the joint information extraction task that finds target mentions together with the sentiment towards each mention from a text corpus. The task is typically modeled as a sequence labeling problem, and solved using state-of-the-art labelers such as CRF. We empirically study the effect of word embeddings and automatic feature combinations on the task by extending a CRF baseline using neural networks, which have demonstrated large potentials for sentiment analysis. Results show that the neural model can give better results by significantly increasing the recall. In addition, we propose a novel integration of neural and discrete features, which combines their relative advantages, leading to significantly higher results compared to both baselines.

1 Introduction

Targeted sentiment analysis has drawn growing research interests over the past few years. Compared with traditional sentiment analysis tasks, which extract the overall sentiment of a document, a sentence or a tweet, targeted sentiment analysis extracts the sentiment over given targeted entities from a text, and therefore is practically more informative. An example is shown in Figure 1. There are at least two practical scenarios:

(1) Certain entities of concern are specified, and the requirement is to extract the sentiment towards their mentions in a text. For example, one can be interested in the sentiment towards *Google Inc.*, *Microsoft* and *Facebook* in financial news texts, or the sentiment towards *Manchester United*, *Liverpool* and *Chelsea* in tweets.

So excited to meet my [**baby Farah**]$_+$!!!
[**Baseball Warehouse**]$_+$: easy to understand information.
The [**#Afghan #Parlaiment Speaker**]$_-$ should Resign .
Saw [**Erykah Badu**]$_-$ last night , vile venue unfortunately .
[**AW service**]$_0$ will be back at work .

Figure 1: *Targeted sentiment analysis*.

(2) No specified target is given, and the requirement is to find sentiments towards entities in the open domain. For example, one might be interested extracting the mentions to all persons and organizations, together with the sentiments towards each mention, from a news archive or a collection of novels.

There are two sub tasks in targeted sentiment analysis, namely entity recognition and sentiment classification for each entity mention which apply to both scenarios above. In scenario (1), *entity recognition* is relatively trivial, and can typically be achieved by pattern matching. Partly due to this reason, most previous work has addressed targeted sentiment analysis as a pure classification task, assuming that target mentions have been given (Jiang et al., 2011; Chen et al., 2012; Dong et al., 2014; Vo and Zhang, 2015). For scenario (2), a named entity recognition (NER) system can be used to extract targets, before the same targeted sentiment classification algorithms are applied. There has also been work that concentrates on extracting opinion targets (Jin et al., 2009; Jakob and Gurevych, 2010). In both cases, the data in Figure 1 can be used for training sentiment classifiers.

Mitchell et al. (2013) took a different approach, extracting named entities and their sentiment classes jointly. They model the joint task

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 612–621,
Lisbon, Portugal, 17-21 September 2015.

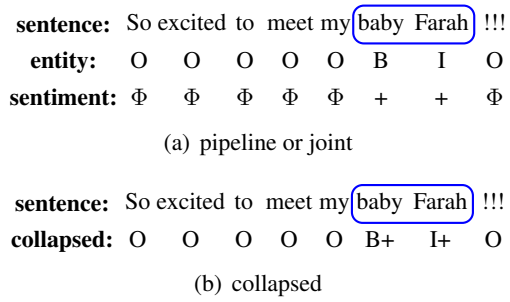


Figure 2: Pipeline, joint and collapsed models for *open targeted sentiment analysis*.

as an extension to the NER task, where an extra sentiment label is assigned to each named entity, in addition to the entity label. As a result, the task can be solved using sequence labeling methods. As claimed by Mitchell et al. (2013), the joint task is particularly suitable when no extra resources are available for training separate syntactic analyzers or name entity recognizers. Such situations can include tweets and low-resource languages/domains. Interestingly, because of containing entity information, the annotation in Figure 1 suffices for training joint entity and sentiment labels even if it is the only resource available.

The annotations in Figure 1 can be transformed into label sequences, as shown in Figure 2. Figure 2 consists of two types of labels, where the B/I/O labels indicate span boundaries, and the +/-/0 labels indicate sentiment classes. The two types of labels can be assigned in a span→sentiment pipeline, or jointly as a multi-label task. Alternatively, as shown in Figure 2(b), the two types of labels can be collapsed into a joint label, such as B+ and I-, indicating the beginning of a positive entity and the middle of a negative entity, respectively. The collapsed labels allow joint entity recognition and sentiment classification to be achieved using a standard sequence labeler.

Mitchell et al. (2013) compare a pipeline model, a joint model and a collapsed model under the same conditional random field (CRF) framework, finding that the pipeline method outperforms the joint model on a tweet dataset. Intuitively, the interaction between entity boundaries and sentiment classes might not be as strong as that between more closely-coupled sources of information, such as word boundaries and POS (Zhang and Clark, 2008), or named entities and constituents (Finkel and Manning, 2009), for which joint models significantly outperform pipeline models. On the other hand, there do exist cases where entity boundaries and sentiment classes reinforce each other. For example, in a tweet such as 'I like *X*.', the contextual pattern indicate both a positive sentiment and an entity in the place of *X*.

Recently, neural network models have been increasingly used for sentiment analysis (Socher et al., 2013; Kalchbrenner et al., 2014; dos Santos and Gatti, 2014), achieving highly competitive results, which show large potentials of neural network models for this task. The main advantages of neural networks are two-fold. First, neural models use real-valued hidden layers to automatically learn feature combinations, which can capture complex semantic information that are difficult to express using traditional discrete manual features. Second, neural networks take distributed word embeddings as inputs, which can be trained from large-scale raw text, thus alleviating the scarcity of annotated data to some extent. In this paper, we exploit structured neural models for open targeted sentiment.

We take the CRF model of Mitchell et al. (2013) as the baseline, and explore two research questions. First, we make an empirical comparison between discrete and neural CRF models, and further combine the strengths of each model via feature integration. Second, we compare the effects of the pipeline, joint and collapsed models for open targeted sentiment analysis under the neural model settings. Our experiments show that the neural model gives competitive results compared with the discrete baseline, with relatively higher recalls. In addition, the integrated model significantly improves over both the discrete and the neural models.

2 Related Work

Targeted sentiment analysis is closely related prior work on aspect-oriented (Hu and Liu, 2004), feature-oriented (Popescu and Etzioni, 2007) and topic-oriented (Yi et al., 2003) sentiment analysis. These related tasks are typically concentrated on product review settings. In contrast, targeted sentiment analysis has a more general setting.

Recently, Wang et al. (2011) proposed a topic-oriented model, which extracts sentiments towards certain topics from tweets. Topics in their model resemble targets in our work, although topics are represented by hashtags, which exists in 14.6% tweets and 27.5% subjective tweets (Wang et al.,

2011). In contrast, targeted sentiment analysis can identify all the mentions to target entities in tweets, thereby having a larger coverage. The drawback is that the identification of mentions is subject to errors, and thus suffers a lower precision compared to hashtag matching.

Sequence labeling models have been used for extracting opinions and target entities as a joint task. Jin et al. (2009) use HMM to extract opinion-baring expressions and opinion targets. Li et al. (2010) improve the results by using CRF to identify the opinion expressions and targets jointly. The task is sometimes referred to as fine-grained sentiment analysis (Wiebe et al., 2005). It is different from our setting in that the predicate-argument relation between opinion-baring expressions and target entities are not explicitly modeled.

Recently, Yang and Cardie (2013) use CRF to extract opinion-baring expressions, opinion holders and opinion targets simultaneously. Their method is also centralized on opinion-baring expressions and therefore in line with Jin et al. (2009) and Li et al. (2010). In contrast, targeted sentiment analysis directly studies entity mentions and the sentiment on each mention, without explicitly modeling the way in which the opinion is expressed. As a result, our task is more useful for applications such as broad-stroke reputation management, but offer less fine-grained operational insight. It requires less fine-grained manual annotation.

As discussed in the introduction, targeted sentiment analysis falls into two main settings. The first is targeted sentiment classification, assuming that entity mentions are given. Most previous work fall under this category (Jiang et al., 2011; Chen et al., 2012; Dong et al., 2014). The second is open domain targeted sentiment, which has been discussed by Mitchell et al. (2013). The task jointly extracts entities and sentiment classes, and is analogous to joint entity and relation extraction (Li and Ji, 2014) in that both are information extraction tasks with multi-label outputs.

Our work is related to the line of work on using neural networks for sentiment analysis. Socher et al. (2011) use recursive auto-encoders for sentiment analysis on the sentence level. They further extend the method to a syntactic treebank annotated with sentiment labels (Socher et al., 2013). More recently, Kalchbrenner et al. (2014) use a dynamic pooling network to include the structure

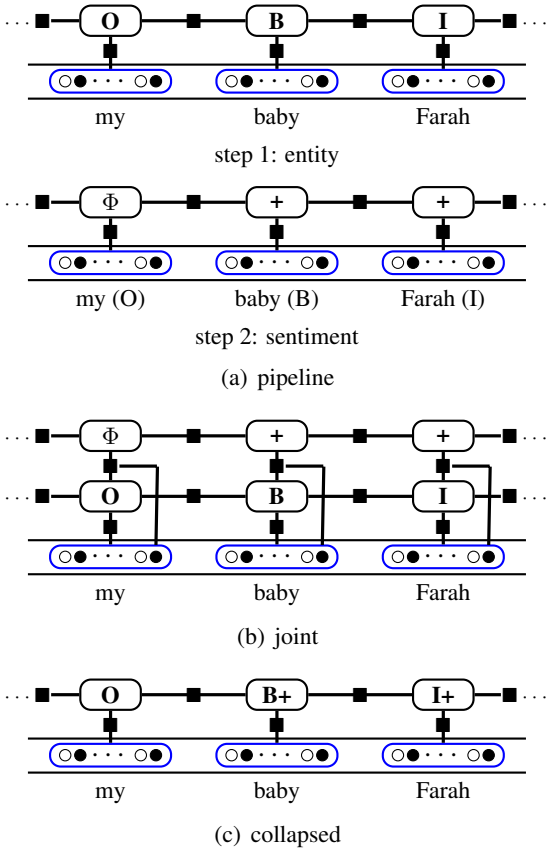


Figure 3: Discrete CRF models for pipeline, joint and collapsed targeted sentiment labeling.

of a sentence automatically, before classifying its sentiment. Zhou et al. (2014) apply deep belief networks for semi-supervised sentiment classification. dos Santos and Gatti (2014) use deep convolution neural networks with rich features to classify sentiments over tweets and movie reviews. These methods use different models to represent sentence structures, performing sentiment analysis on the sentence level, without modeling targets.

Dong et al. (2014) perform targeted sentiment classification by using a recursive neural network to model the transmission of sentiment signal from opinion baring expressions to a target. They assume that the target mention is given, and perform three-way sentiment classification. In contrast, we apply a structural neural model for open domain targeted sentiment analysis, identifying and classifying all targets in a sentence simultaneously.

3 Discrete CRF Baselines

As shown in Figure 2, the input $\vec{x}$ to our tasks is a word sequence. Assuming no external resources, there is no POS given to each input word x_i. For

the pipeline and collapsed tasks, there is a single output label sequence $\vec{y}$. For the joint task, there are two label sequences $\vec{y}$ and $\vec{z}$, for entity and sentiment labels, respectively. We take the models of Mitchell et al. (2013) as our baseline, which are standard CRFs with discrete manual features. To facilitate comparison between the discrete baseline and our neural models, we give a unified formulation to all the models in this paper, introducing the neural and integrated models as extensions to the discrete models.

The baseline CRF structures for pipeline, joint and collapsed targeted sentiment analysis are shown in Figure 3(a), 3(b) and 3(c), respectively. In the figures, the input features are represented as black and white circles, indicating that they take $0/1$ binary values. The labels O, B and I indicate a non-target, the beginning of a target, and part of a target, respectively. The labels $+$, $-$, 0 and Φ indicate positive, negative, neutral and *NULL* sentiments, respectively. The *NULL* sentiment is assigned to O entities automatically, and modeled as a hidden variable in the pipeline and joint CRFs.[1] The collapsed labels take combined meanings from their components.

The links between labels and inputs represent output clique potentials:

$$\Psi(\vec{x}, y_i) = exp\Big\{\vec{\theta} \cdot \vec{f}(\vec{x}, y_i)\Big\},$$

where $\vec{f}(\vec{x}, y_i)$, is a discrete manual feature vector, and $\vec{\theta}$ is the model parameter vector.

The links between labels represent edge clique potentials:

$$\Phi(\vec{x}, y_i, y_{i-1}) = exp\Big\{\tau(y_i, y_{i-1})\Big\},$$

where $\tau(y_i, y_{i-1})$ is the transition weight, which is also a model parameter.

For both the pipeline and collapsed models, the conditional probability of a label sequence given an input sequence is:

$$P(\vec{y}|\vec{x}) = \frac{\prod_{i=1}^{|x|} \Psi(\vec{x}, y_i) \prod_{j=1}^{|x|} \Phi(\vec{x}, y_i, y_{i-1})}{Z(\vec{x})},$$

[1]Note the difference between neural and *NULL* sentiments. The former indicates that a target does not bare any sentiment, and the latter simply means that the term is not a part of a target.

surface features
word identity; word length; message length;
punctuation characters; has digit; has dash; is lower case;
is 3 or 4 letters; first letter capitalized; sentence position;
more than one letter capitalized; Jerboa features;
linguistic features
function words; can syllabify; curse words;
laugh words; words for good, bad, no, my;
intensifiers; slang words; abbreviations;
common verb endings; common noun endings;
subjective suffixes and prefixes;
cluster features
Brown cluster at length 3; Brown cluster at length 5;
sentiment features
is sentiment-bearing word; prior sentiment polarity;

Table 1: Discrete features.

where $Z(\vec{x})$ is the partition function:

$$Z(\vec{x}) = \sum_{\vec{y}'} \Big(\prod_{i=1}^{|x|} \Psi(\vec{x}, y'_i) \prod_{j=1}^{|x|} \Phi(\vec{x}, y'_i, y'_{i-1}) \Big),$$

For the joint model, we apply a multi-label CRF structure, where there are two separate sets of output clique potentials $\Psi_1(\vec{x}, y_i)$ and $\Psi_2(\vec{x}, z_i)$ and two separate sets of edge clique potentials $\Phi_1(\vec{x}, y_i, y_{i-1})$ and $\Phi_2(\vec{x}, z_i, z_{i-1})$ for the label sets $\{B, I, O\}$ and $\{+, -, 0\}$, respectively. In the Figure 3(b), there are also links between the span label y_i and the sentiment label z_i for each word x_i. These links indicate label dependencies, which are constraints for decoding. For example, if $y_i = O$, then z_i must be ϕ.

We apply Viterbi decoding for all tasks, and training is performed using a max-margin objective, which is discussed in Section 6. Our training algorithm is different from that of Mitchell et al. (2013), but gives similar discrete CRF accuracies in our experiments. Wang and Mori (2009) also applied a max-margin trainig strategy to train CRF models. The set of features is taken from Mitchell et al. (2013) without changes, as shown in Table 1. Here the cluster features refer to Brown word clusters (Brown et al., 1992).

4 Neural Models

We extend the discrete baseline system with two salient changes, which are illustrated in Figure 4. First, the input discrete features are replaced with continuous word embeddings. Each node in the input takes a real value between 0 and 1, as represented by grey nodes in Figure 4. Second, a hidden

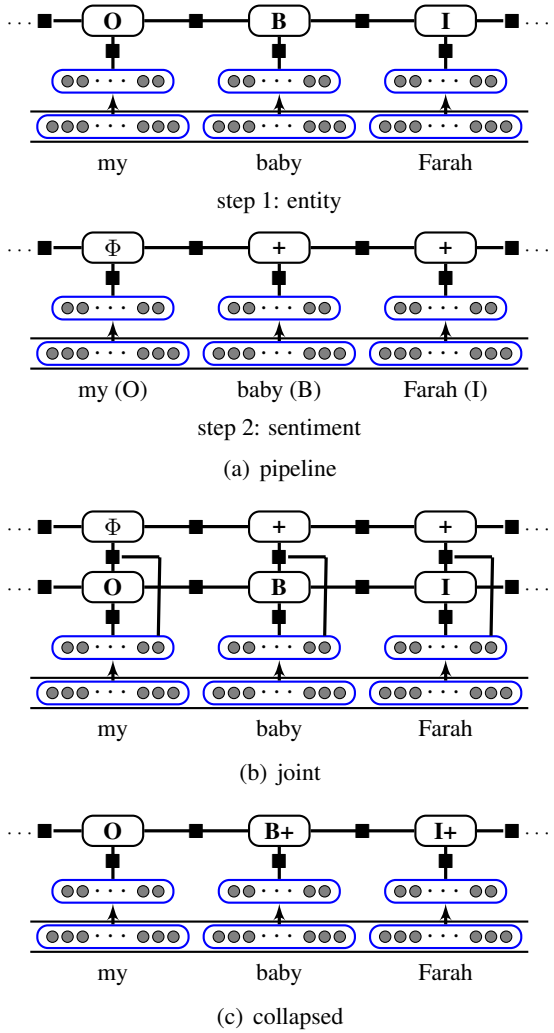


Figure 4: Neural networks for pipeline, joint and collapsed targeted sentiment labeling.

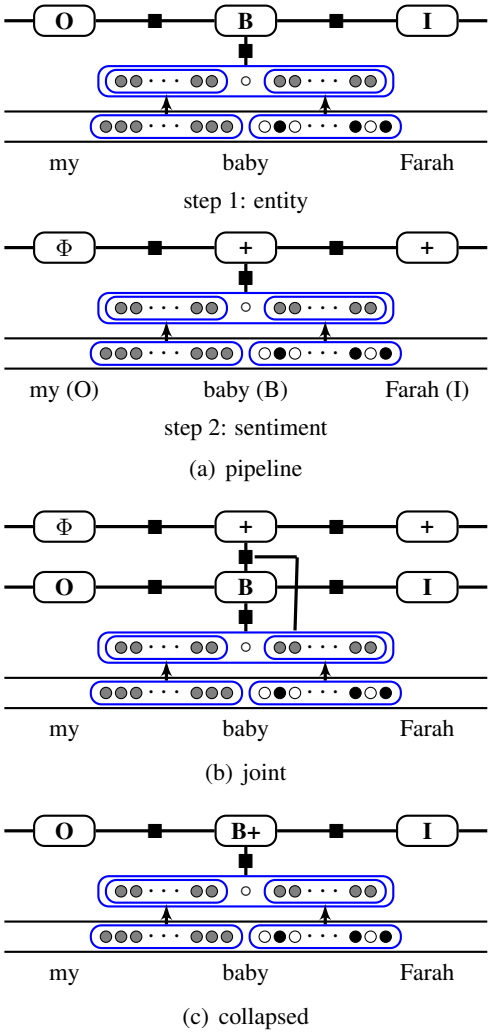


Figure 5: Integrated models for pipeline, joint and collapsed targeted sentiment labeling.

neural layer $\vec{h}$ is added between the input nodes $\vec{x}$ and the label nodes y_i.

Formally, the links between the input nodes $\vec{x}$ and the hidden nodes $\vec{h}_i$ for the node y_i in Figure 4 represent a feature combination function:

$$\vec{h}_i = tanh\Big(\mathbf{W}.(e(\vec{x}_{i-2}) \oplus e(\vec{x}_{i-1}) \oplus e(\vec{x}_i) \oplus e(\vec{x}_{i+1}) \oplus e(\vec{x}_{i+2})) + \vec{b}\Big)$$

where e is the embedding lookup function, $\oplus$ is the vector concatenation function, the matrix $\mathbf{W}$ and vector $\vec{b}$ are model parameters and *tanh* is the activation function.

The output clique potential of y_i becomes:

$$\Psi(\vec{x}, y_i) = exp\Big\{\vec{\sigma} \cdot \vec{h}_i\Big\}$$

where $\vec{\sigma}$ is a model parameter, and the edge clique potentials remain the same as the baseline. By using a hidden layer for automatic feature combinations, the neural model is free of manual features, and can benefit from unsupervised embeddings. Decoding and training are performed using the same algorithms as the baseline.

The major neural architectures in Figure 4 have been explored as *conditional neural fields* by Peng et al. (2009) and *neural conditional random fields* by Do et al. (2010), and is connected to the sentence-level likelihood neural networks of Collobert et al. (2011), as pointed out by Wang and Manning (2013b). The main differences between our model and the prior work are in the multi-label settings and training details.

5 Integrated Models

Gleaning different sources of information, neural features and discrete linear features comple-

ments each other. As a result, a model that integrates both features can potentially achieve performance improvements. Most work attempts to add neural word embeddings into a discrete linear model (Turian et al., 2010; Yu et al., 2013; Guo et al., 2014), or add discreted features into a neural model (Ma et al., 2014). We make a novel combination of the discrete models and the neural models by integrating both types of inputs into a same CRF framework.[2]

The architectures of the integrated models are shown in Figure 5. The main difference between Figure 5 and Figure 3 is the input layer. The integrated model takes both continuous word embeddings, which are shown in grey nodes, and discrete manual features, which are shown in black or white nodes, as the input.

A separate hidden layer is given to each type of input nodes, with the hidden layer for the embeddings being the same as the neural baseline:

$$\vec{h}_i = tanh\Big(\mathbf{W} \cdot (e(\vec{x}_{i-2}) \oplus e(\vec{x}_{i-1}) \oplus e(\vec{x}_i) \oplus e(\vec{x}_{i+1}) \oplus e(\vec{x}_{i+2})) + \vec{b}\Big)$$

The hidden nodes $\vec{g}_i$ between the discrete features and the node y_i are:

$$\vec{g}_i = tanh\Big(\vec{\theta} \cdot \vec{f}(\vec{x}, y_i)\Big)$$

Finally, the output clique potential of y_i becomes:

$$\vec{\Psi}(\vec{x}, y_i) = exp\Big\{\vec{\sigma} \cdot (\vec{h}_i \oplus \vec{g}_i)\Big\}$$

The edge clique potentials remain the same as the baseline models; the same training and decoding algorithms are used.

6 Training

We use a max-margin objective to train our model parameters Θ, which consist of $\vec{\theta}$, τ, $\mathbf{W}$, $\vec{b}$ and $\vec{\sigma}$ for each model. The objective function is defined as:

$$L(\Theta) = \frac{1}{N}\sum_{n=1}^{N} l(\vec{x}_n, \vec{y}_n, \Theta) + \frac{\lambda}{2} \parallel \Theta \parallel^2,$$

where $(\vec{x}_n, \vec{y}_n)|_{n=1}^{N}$ are the set of training examples, λ is a regularization parameter, and $l(\vec{x}_n, \vec{y}_n, \Theta)$ is the loss function towards one example $(\vec{x}_n, \vec{y}_n)$.

The loss function is defined as:

$$l(\vec{x}_n, \vec{y}_n, \Theta) = \max_{\vec{y}}(s(\vec{x}_n, \vec{y}, \Theta) + \delta(\vec{y}, \vec{y}_n)) - s(\vec{x}_n, \vec{y}_n, \Theta),$$

where $s(\vec{x}, \vec{y}, \Theta) = logP(\vec{y}|\vec{x})$ is the log probability of $\vec{y}$, and $\delta(\vec{y}, \vec{y}_n)$ is the Hamming distance between $\vec{y}$ and $\vec{y}_n$.

We use online learning to train model parameters, updating the parameters using the AdaGrad algorithm (Duchi et al., 2011). One thing to note is that, our objective function is not differentiable because of the loss function $l(\vec{x}_n, \vec{y}_n, \Theta)$. Thus we use sub-gradients for $l(\vec{x}_n, \vec{y}_n, \Theta)$ instead, which can be computed by the formula:

$$\frac{\partial l(\vec{x}_n, \vec{y}_n, \Theta)}{\partial \Theta} = \frac{\partial s(\vec{x}_n, \hat{\vec{y}}, \Theta)}{\partial \Theta} - \frac{\partial s(\vec{x}_n, \vec{y}_n, \Theta)}{\partial \Theta},$$

where $\hat{\vec{y}}$ is the predicted label sequence which corresponds to $l(\vec{x}_n, \vec{y}_n, \Theta)$.

Maximum-likelihood training is a commonly used alternative to max-margin training for neural networks. It has been applied to the models of Do et al. (2010) and Collobert et al. (2011), for example. However, our experiments show that maximum-likelihood training cannot be applied to open-domain targeted sentiment tasks. Although giving comparable overall accuracies in both entity and sentiment labels, it suffers from unbalanced sentiment labels, assigning the neutral sentiment to most entities. This problem can be addressed by imposing a polarity-sensitive cost to the training, such as the sentence-level averaged F1-score between positive, negative and neutral labels. We skip these results due to space limitations. In contrast, max-margin training does not suffer from the label skew issue, thanks to the use of Hamming loss in the objective function.

7 Experiments

7.1 Experimental Settings

Data: We use the data of Mitchell et al. (2013)[3] to conduct all the experiments, which consist of entity and sentiment annotations on both English and Spanish tweets. Simple normalizations are

[2] Wang and Manning (2013a) also investigated the integration of discrete and neural features in CRF models. They compared the effect of integration without hidden layers (i.e. Turian et al. (2010)) and with hidden layers (i.e. our methods) for NER and chunking, finding that the formal outperforms the latter. Our results are different from theirs, and a hidden layer gives significant improvements to the targeted sentiment analysis task.

[3] http://www.m-mitchell.com/code/index.html

Domain	#Sent	#Entities	#+	#-	#0
English	2,350	3,288	707	275	2,306
Spanish	5,145	6,658	1,555	1,007	4,096

Table 2: Experimental corpus statistics.

conducted to replace all usernames and URLs into the special tokens $\langle username \rangle$ and $\langle url \rangle$, respectively. Following Mitchell et al. (2013), we report ten-fold cross-validation results. During training, we split 10% of the training corpus as the development corpus to tune hyper-parameters. Table 2 shows the corpus statistics.

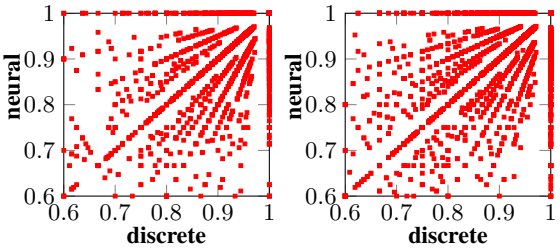
Parameters: For all the neural models, we set the hidden layer size $|\vec{h}|$ for neural features to 200, the hidden layer size $|\vec{g}|$ for discrete features to 30, the initial learning rate for adagrad to 0.01 and the regularization parameter λ to 10^{-8}. English and Spanish word embeddings are trained using the *word2vec* tool[4], with respective corpora of 20 minion random tweets crawled by tweet API[5]. The size of word embeddings is 100. For English, there are 8,061 unique words, for which 25% are out of *word embedding* vocabulary (OOE) words, while for Spanish, there are 14,648 unique words, for which 15% are OOE words.

Metrics: We take full-span metrics for evaluation, which is different from Mitchell et al. (2013), who evaluate mainly the beginning of spans. We measure the precision, recall and F-score of entity recognition (**Entity**), targeted sentiment analysis (**SA**) (both entity and sentiment), and targeted subjectivity detection (**Subjectivity**) (both entity and subjectivity, namely merging the + and - labels as "1" label, and performing two-way 0/1 subjectivity classification on entities). For **SA**, an entity is taken as correct only when the span and the sentiment are both correctly recognized. Similarly, for **Subjectivity**, an entity is taken as correct only when both the span and the subjectivity are correctly recognized.

Code: We make the C++ implementations of the discrete, neural and combined models available and GPL, at `https://github.com/SUTDNLP/OpenTargetedSentiment`.

7.2 Comparing Neural and Discrete Models

The main results on both the English and Spanish dataset are shown in Table 3, which are measured on the pipeline, the joint and the collapsed tasks, respectively. As can be seen from the table, the neural models give higher F-scores than the discrete CRF models on the English dataset, while comparable overall F-scores on the Spanish dataset. The gains on English are mostly attributed to improved recalls, while the precision of the neural CRF models are relatively lower. A likely reason for this observation is that the neural model takes embedding inputs, which allow semantically similar words to be represented with similar vectors. As a result, the neural model can better capture patterns that do not occur in the training data. In contrast, the discrete model is based on manually defined binary features, which do not fire if not contained in the training data. Because discrete feature instantiation is based on exact matching, the discrete model gives a relatively higher precision.



(a) English (b) Spanish

Figure 6: Labeling accuracy comparisons.

To further contrast the discrete and neural models, we draw the per-word accuracies of sentiment labels according to both models in Figure 6. In the figure, each dot represents the accuracy of a sentence, measured in the pipeline task. The dots for both English and Spanish are scattered from the reverse diagonal, showing that the two models make very different errors, which suggests that model integration can lead to better accuracies.

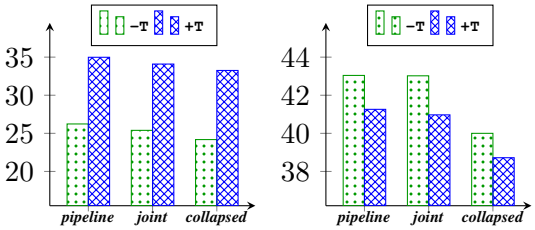
7.3 The Integrated Model

As shown in Table 3, the integrated model combines the relative advantages of both pure models, improving the recall over the discrete model and the precision over the neural model. In most cases, it gives the best results in terms of both precision and recall. For the English pipeline model, the integrated model improves the entity recognition F-score from 43.84% to 55.67% (significant with $p < 10^{-5}$ by pair-wise t-test) as compared to the discrete baseline, namely Mitchell et al. (2013).

[4]https://code.google.com/p/word2vec/
[5]https://dev.twitter.com/

Model	English						Spanish					
	Entity			SA			Entity			SA		
	P	R	F	P	R	F	P	R	F	P	R	F
Pipeline												
discrete	59.37	34.83	43.84	42.97	25.21	31.73	**70.77**	47.75	57.00	**46.55**	31.38	37.47
neural	53.64	44.87	48.67	37.53	31.38	34.04	65.59	47.82	55.27	41.50	30.27	34.98
integrated	**60.69**	**51.63**	**55.67**	**43.71**	**37.12**	**40.06**	70.23	**62.00**	**65.76**	45.99	**40.57**	**43.04**
Joint												
discrete	59.55	34.06	43.30	43.09	24.67	31.35	71.08	47.56	56.96	46.36	31.02	37.15
neural	54.45	42.12	47.17	37.55	28.95	32.45	65.05	47.79	55.07	40.28	29.58	34.09
integrated	**61.47**	**49.28**	**54.59**	**44.62**	**35.84**	**39.67**	**71.32**	**61.11**	**65.74**	**46.67**	**39.99**	**43.02**
Collapsed												
discrete	**64.16**	26.03	36.95	**48.35**	19.64	27.86	73.18	35.11	47.42	**49.85**	23.91	32.30
neural	58.53	37.25	45.30	43.12	27.44	33.36	67.43	43.2	52.64	42.61	27.27	33.25
integrated	63.55	**44.98**	**52.58**	46.32	**32.84**	**38.36**	**73.51**	**53.3**	**61.71**	47.69	**34.53**	**40.00**

Table 3: Main results.



(a) Neural (b) Integrated

Figure 7: Effect of fine-tuning (+T — with fine-tuning; -T — without fine-tuning).

The overall SA score is improved from 31.73% to 40.06% ($p < 10^{-5}$). Similar improvements are achieved to the other test datasets.

7.4 Fine-tuning Word Embeddings

In the experiments above, word embeddings are fine-tuned for the neural models, but not for the integrated models. By fine-tuning, embeddings of in-vocabulary words are treated as model parameters, and updated with other parameters in supervised training. This can improve the accuracy of the model by significantly enlarging the parameter space. However, it can make the embeddings of OOV words less useful to the model, because the hidden layers are tuned with adjusted embeddings.

Figure 7 shows the effectiveness of fine-tuning on the neural and integrated models using the Spanish data. Similar findings apply to the English data. The neural model heavily relies on fine-tuning of embeddings, and a likely reason is that manual discrete features offer sufficient parameters for capturing in-vocabulary patterns. On the other hand, thanks to the rich discrete features in parameter space, the integrated model does not rely on fine-tuning of word embeddings, which even caused slight overfitting and reduced the performances. This makes the non-fine-tuned integrated model potentially advantageous in handling test data with many OOV words.

7.5 Comparing pipeline, joint and collapsed models

Mitchell et al. (2013) find that for discrete CRF, the pipeline task gives competitive overall performances compared with the joint task. This suggests a relatively weak connection between entity boundary information and sentiment classes. We re-examine the comparisons under the neural network setting, where automatic feature combinations can be useful in capturing more subtle correlations between two sources of information.

As shown in Table 3, the overall results are similar to those of Mitchell et al. (2013), with both the neural and the integrated models demonstrating the same trends as the discrete baselines. A more detail analysis, however, shows some relative strengths of the joint task. Table 4 give the precision, recall and F-scores of subjectivity, and those of SA excluding neutral sentiment labels on the Spanish data. Findings on the English dataset are consistent.

The latter metrics highlight sentiment polarities, which can be relatively more useful. The joint task gives better F-scores on both metrics, which suggest that is a considerable choice for open targeted sentiment. When there is external resource for en-

Model	Subjectivity			SA/0		
	P	R	F	P	R	F
pipeline	47.92	**42.26**	44.84	**42.93**	18.02	25.14
joint	49.17	42.13	**45.32**	40.93	**21.62**	**27.93**
collapsed	**49.63**	35.94	41.63	42.10	15.62	22.49

Table 4: Results on subjectivity and polarity.

tity recognition, the pipeline can be a favorable choice. On the other hand, although useful for some joint sequence labeling task (Ng and Low, 2004), the collapsed task does not seem to address the joint sentiment task as effectively. We find this result empirical, but consistent across our datasets.

8 Conclusion

We explored open domain targeted sentiment analysis using neural network models, which gave competitive results when evaluated against a strong discrete CRF baseline, with relatively higher recalls. Given complementary error distributions by the discrete and neural CRFs, we proposed a novel combination which significantly outperformed both models. Under the neural setting, we find that it is preferable to solve open targeted sentiment as a pipeline or joint multi-label task, but not as a joint task with collapsed labels.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, which helped to improve the paper. This work is supported by the Singapore Ministry of Education (MOE) AcRF Tier 2 grant T2MOE201301 and SRG ISTD 2012 038 from Singapore University of Technology and Design.

References

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Lu Chen, Wenbo Wang, Meenakshi Nagarajan, Shaojun Wang, and Amit P Sheth. 2012. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *ICWSM*.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Trinh Do, Thierry Arti, et al. 2010. Neural conditional random fields. In *International Conference on Artificial Intelligence and Statistics*, pages 177–184.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Jenny Rose Finkel and Christopher D Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 110–120.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1035–1045.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 151–160.

Wei Jin, Hung Hay Ho, and Rohini K Srihari. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 465–472.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the Association for Computational Linguistics*.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 653–661.

Ji Ma, Yue Zhang, and Jingbo Zhu. 2014. Tagging the web: Building a robust web tagger with neural network. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 144–154.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *EMNLP*, pages 277–284.

Jian Peng, Liefeng Bo, and Jinbo Xu. 2009. Conditional neural fields. In *Advances in neural information processing systems*, pages 1419–1427.

Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642.

Joseph Turian, Lev-Arie 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.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1347–1353.

Mengqiu Wang and Christopher D. Manning. 2013a. Effect of non-linear deep architecture in sequence labeling. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1285–1291.

Sida Wang and Christopher Manning. 2013b. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 118–126.

Yang Wang and Greg Mori. 2009. Max-margin hidden conditional random fields for human action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 872–879.

Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. 2011. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1031–1040.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *ACL (1)*, pages 1640–1649.

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 427–434.

Mo Yu, Tiejun Zhao, Daxiang Dong, Hao Tian, and Dianhai Yu. 2013. Compound embedding features for semi-supervised learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 563–568.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL-08: HLT*, pages 888–896.

Shusen Zhou, Qingcai Chen, Xiaolong Wang, and Xiaoling Li. 2014. Hybrid deep belief networks for semi-supervised sentiment classification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1341–1349.