ELS: A Word-Level Method for Entity-Level Sentiment Analysis

Nikos Engonopoulos
Department of Informatics and Telecommunications
University of Athens
Greece

Angeliki Lazaridou
Department of Informatics and Telecommunications
University of Athens
Greece

Georgios Paliouras
Institute of Informatics and Telecommunications
NCSR "Demokritos"
Greece
paliourg@iit.demokritos.gr

Konstantinos Chandrinos
i-sieve technologies Ltd.
NCSR "Demokritos"
Greece

ABSTRACT

We introduce ELS, a new method for entity-level sentiment classification using sequence modeling by Conditional Random Fields (CRF). The CRF is trained to identify the sentiment of each word in a document, which is then used to determine the sentiment for the entity, based on where it appears in the text. Due to its sequential nature, the CRF classifier performs better than the common bag-of-words approaches, especially when we target the local sentiment in small parts of a larger document. Identifying the sentiment about a specific entity, mentioned in a blog post or a larger product review, is a special case of such local sentiment classification. Furthermore, the proposed approach performs well even in short pieces of text, where bag-of-words approaches usually fail, due to the sparseness of the resulting feature vector. We have implemented and tested the proposed method on a publicly available benchmark corpus of short product reviews in English. The results that we present in this paper improve significantly upon published results on the same data, thus confirming our intuition about the approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation, Performance

WIMS'11 May 25-27, 2011 Sogndal, Norway
Copyright 2011 ACM 978-1-4503-0148-0/11/05 ...$10.00.

Keywords

Sentiment Classification, Opinion Mining, Fine-grained sentiment analysis, Statistical Sequence Modeling, Pattern Discovery

1. INTRODUCTION

"What if you could quickly discover, quantify and act on the opinions of your customers and influencers wherever they appeared?"[1]

Sentiment analysis is defined [9] as the extraction of information, concerning sentiment expressed by people in textual data (reviews, blogs, fora etc). One particular case of sentiment analysis that has recently gained significant commercial and research interest is the identification of sentiment towards specific entities, e.g. products, such as MP3 players, and their features, such as battery, screen etc. This process is automated by text analysis aiming at a reasonably unbiased estimate of the opinion that people hold about products, brands, etc. It is considered more valuable than traditional market research tools, as it usually has a stronger statistical basis, it can be conducted across the world and it is unobtrusive, in the sense that the subjects are not explicitly prompted for their opinion.

The terms *sentiment*, *opinion* and *subjectivity* are used interchangeably in the literature, often causing confusion. In this paper, we will use the term *sentiment* for the three-valued label (positive, negative, neutral) assigned to a part of a text that describes *an attitude, thought, or judgment prompted by feeling*[2] towards an entity. The terms *opinion* and *subjectivity* will be used to refer to non-neutral sentiment (positive or negative), i.e. a *subjective* phrase is defined as one that contains opinion. Therefore, sentiment classification is a three-valued classification task into positive, negative, neutral, while opinion extraction is a binary classification task into subjective, objective, with the objective class representing neutral sentiment and the subjective class representing positive or negative sentiment. Once a phrase is found to contain an opinion, the *polarity* or sentiment orientation of that opinion is defined as the sentiment conveyed by the phrase – by definition it will be either positive or negative.

Research on sentiment analysis so far has concentrated on extracting information from text using either rule-based natural lan-

[1] Sentiment Analysis: Drive Business Agility by Quantifying What People Think, `http://bit.ly/9Bpc6v`

[2] Online Merriam-Webster Dictionary: `http://www.merriam-webster.com/dictionary/sentiment`

guage processing (NLP) or statistical information retrieval (IR) techniques. Rule-based NLP approaches usually utilise linguistic information and/or reasoning to infer the sentiment expressed towards a tagret. However, they are either too generic and thus inappropriate for a specific domain/ market or too complex and not easily scalable to large datasets, requiring domain-specific linguistic resources, such as grammars, parsers and lexica [6]. In particular, approaches that use subjectivity or sentiment orientation (SO) lexica face the problem that the same word may convey different sentiments in different contexts [2]. It is evident, for instance, that in the phrases *"I badly want it"* and *"it is badly manufactured"* the adverb "badly" conveys two totally different sentiments for the entity "it". For this reason, the effectiveness of subjectivity or SO labels for individual words is limited. On the other hand, IR approaches typically use statistical and machine learning methods to train bag-of-words classifiers, such as SVM, Naive Bayes etc., which ignore the order of words within a text. This can prove ineffective, especially when considering small extracts of text, where word order can determine the sentiment. For instance, the sentence

> "My old MP3 player has better sound quality than my brand-new Creative"

has the same bag-of-words representation as

> "My brand-new Creative has better sound quality than my old MP3 player".

Entity-level sentiment analysis is particularly prone to this problem, as the sentiment to be identified is expressed very locally in the text. Typical cases are blog posts, where the author expresses an opinion about a product, among many other things, or large product comparison articles, where the product that we are interested in is one among many while receiving different sentiment each.

ELS goes some way towards solving the problem, by identifying the sentiment of each word in the document, taking into account the order in which words appear. For this purpose, we use the sequential classification method of Conditional Random Fields (CRF) [5], which has become very popular for information extraction from text, due to its scalability and very competitive performance. We tested the proposed method on a publicly available benchmark corpus of short product reviews[3]. Our results confirm our initial intuition, as ELS achieves significantly higher performance than that reported in the literature [3].

The rest of the paper is structured as follows: section 2 provides a brief overview of work on entity-level sentiment analysis, section 3 describes the proposed method and the CRF model that we used, section 4 presents our experimental results and compares them with the literature, while section 5 summarizes the main contributions of this work and highlights possible improvement paths.

2. RELATED WORK

Several different methods have been proposed in the literature for sentiment analysis, addressing the issue at various levels of granularity. Some researchers have proposed methods for document-level sentiment classification (e.g. [10], [15]). At this high level of granularity, it is often impossible to infer the sentiment expressed about particular entities that are mentioned in the text, as a document may convey different opinions for different entities. Thus when it comes to mining opinions about entities, such as products in product reviews, it has been shown that sentence- and phrase–level ([14], [17] , [18], [16]) analysis lead to a performance gain. Sometimes, even at those levels, one may have to deal with phrases or sentences which include references to more than one entity with conflicting sentiments for example:

> "On the other hand, Canon flushed compatibility down the toilet in 1985 when it created a new and completely incompatible system of AF cameras and lenses called EOS. Nothing works together before or after the great divide of 1985.
>
> To Canon's credit, the new EOS system is a better design than the old Nikon mount, but old Canon FD manual focus lenses, once promoted as "timeless" by Canon, are useless today on modern Canon cameras. Contrast this to Nikon, where just about every lens ever made works swell, with few limitations, on every brand new camera.
>
> Every Canon AF lens works on every Canon AF camera, including the digital SLRs, except for Canon's EF-S lenses, which only work on some of the newest 1.6x cameras. 1980's Canon AF lenses work great on every current Canon camera. "

Popescu and Etzioni [11] addressed the problem of entity-level classification by creating OPINE, an information extraction system which uses extraction rules to retrieve sentences that contain opinion towards known features or entities. OPINE uses the semantic orientation of words and other linguistic information, in order to determine the sentiment orientation (SO) of those words in the context of particular opinion sentences. This approach goes some way towards addressing the issue of the same word having different sentiments in different contexts. However, it is limited in that it depends on syntactic analysis rules - which may be language-specific - in determining which sentences contain opinion towards entities. Similarly in [3], opinion sentences are also extracted first and then the polarity of the sentiment towards known entities is identified, using known polar adjectives obtained from WordNet.

Another approach that deals only with the problem of sentiment orientation is introduced by Liu and Ding [2]. Their rule-based method uses a lexicon consisting of words having a sentiment label (positive, negative or context-dependent), together with a number of linguistic rules used to infer the sentiment label of the context-dependent words. They segment each sentence using BUT words as delimiters (i.e. "but", "except that", etc) and then use a simple scoring function to decide the sentiment conveyed in each segment. However, BUT words not always entail a sentiment alternation and they are not the only indicators of a change in sentiment inside a sentence (e.g. in the sentence "Although this camera takes great pictures, has a short battery life", the word "Although" does not indicate a change in the sentiment between its left and right context; this change occurs later in the sentence). This method is shown to outperform those presented in [3] and [11].

In our method, we opted for finer-grained word-level sentiment classification. The main motivation behind this was that the task of extracting sentiment for specific entities heavily depends on local context. The proposed method assigns sentiment labels to entities, based on the sentiment of the segment in which the entity appears. In this manner, the annotation remains at a low enough level of granularity (text segments of arbitrary length) that permits the learned model to capture changes in sentiment that occur in the smallest possible text segment. Furthermore, our method does not make use of any sentiment lexicon and does not perform complex syntactic analysis. It is, therefore, a significant step towards language and domain independence, as it does not require significant prior knowledge about the domain or the language of the data.

[3] Customer Review Datasets, `http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`

The resulting sequence of sentiment labels for each word provides a rich source of information, through which several sentiment-related phenomena can be observed and analyzed. The details of our method are discussed in the following sections.

In order to perform sequence labeling we use linear-chain Conditional Random Fields, a probabilistic model which was first presented by McCallum and Sutton [5]. Sharifi & Cohen [13] have already indicated the potential of using CRF for document-level and sentence-level sentiment classification, by extracting domain-specific polar words from text. Furthermore, Mao & Lebanon [7] have used CRF for sentence-level sentiment classification, treating sentiment as an ordinal, instead of a categorical variable. For this purpose, they have incorporated a set of monotonicity constraints into the model (isotonic CRF). Zhao, Liu & Wang [19] also used CRF for sentence-level sentiment classification, by modifying the label set into a three-layer hierarchy of labels (subjectivity, polarity and sentimental strength). In this approach, any inconsistencies that are observed between layers help avoiding error propagation. Sadamitsu, Sekine & Yamamoto [12] addressed the problem of sentence-level sentiment classification by using as features for their CRF classifier (Hidden CRF) words from sentences that reverse the meaning of sentences (i.e. but, not). CRF has also been used for extracting opinion targets [4]; however, in this paper we consider that opinion targets are already known. Nakagawa, Inui and Kurohashi [8] also trained CRF on dependency-parsed subjective sentences, using the sentiment polarity of the intermediate nodes of the dependency tree as hidden variables; the task was again to identify the sentiment polarity of whole sentences.

As indicated by the examples above, sentence-level classification is not sufficient for identifying the sentiment expressed about particular entities. In this direction, Breck et al. [1] use CRF to extract opinions from the MPQA corpus [17], in the form of either direct subjective expressions (DSE) or expressive subjective elements (ESE). This is similar to the task of detecting subjectivity at the segment level in our method. The results of their approach improve previous experimental results on the same corpus, further demonstrating that CRF is an appropriate choice for sentiment analysis at a fine-grained level. In this work, we additionally propose the use of CRF for sentiment polarity labeling at the level of individual words.

3. WORD-LEVEL SENTIMENT ANALYSIS

In this section we present the new method for word-level sentiment analysis. We start, in section 3.1, by presenting the underlying philosophy of the method and then, in section 3.2, we present the sequential classification model and the way in which it is trained. Finally, in section 3.3, we explain how the sequence of word labels can be used to identify the sentiment expressed for specific entities.

3.1 Word-level sentiment classification

The main motivation behind our choice of performing word-level sentiment classification is that, at such a low level of granularity, text refers to one entity and expresses a single sentiment towards that entity. The human reader usually knows the exact sentiment expressed for an *entity* that is referred to at a given point of discourse. Therefore, we attempt to capture the *sentiment flow* that leads to the particular sentiment expressed about an entity.

Let X be a random variable over data sequences to be labeled and Y a random variable over corresponding label sequences. The components y_i of Y are assumed to range over a finite label alphabet, in our case *positive*, *negative*, *neutral*. We define the notion of sentiment flow as being the sequence of sentiment labels $Y =< y_1, y_2, ..., y_k >$ that corresponds to a sequence of words $X =< x_1, x_2, ..., x_k >$. x_i correspond to natural language words and other tokens found in text, while y_i to labels from the restricted set. Using again the example presented in section 1, the sentiment flow leading to the entity "my brand-new Creative" in the sentence

> "My old MP3 player has better sound quality than my brand-new Creative"

is identical to that for "my old MP3" in the sentence

> "My brand-new Creative has better sound quality than my old MP3 player".

For the purposes of our experiments, we do not make a distinction between products and product features and we consider them all as entities, e.g. given the sentence

> "Creative is an excellent mp3 player, although expensive, but its supplied earphones are of inferior quality",

"Creative mp3 player" as well as "price of Creative mp3 player" and "earphones of Creative mp3 player" are all considered different entities. A given *sentence*, i.e. a sequence of *words* including punctuation marks delimited by a full stop, may contain different sentiments for different entities. However, a sentence can be split into smaller parts (*segments*), each of which refers *at most* to one entity, and conveys *a single* sentiment (or none) towards it. Thus, the segment contains a sequence of words, which is at least as long as the name of the entity or a pronoun / adjective which directly refers to the entity (*entity reference*).

Table 1 presents the segments of the example used above. This example sentence is divided into three segments, separated by rectangles, and in each segment there is one entity reference. It thus becomes clear that each segment conveys a unique sentiment for the entity that it contains:

- positive sentiment towards *Creative mp3 player*
- negative sentiment towards *price of Creative mp3 player*
- negative sentiment towards *earphones of Creative mp3 player*

Based on this definition of a segment, we can assign the sentiment of the segment to each of the words that it contains, and model word-level sentiment classification as a sequential labeling problem. This is a similar approach to that used for information extraction using CRF, where each word of a named entity takes the label (type) of the entity. Here, we extend this approach to the notion of segments, as defined above.

3.2 Training word-level classifiers

Our approach to sequential sentiment labeling is based on the use of Conditional Random Fields (CRF).[4] CRF are a discriminative approach to sequence labeling, which, in contrast to their generative counterparts (Hidden Markov Models), scales well to large sets of features and provides usually more accurate classification. They are thus preferred when the problem can be modeled as a sequence classification task. Figure 1 illustrates the structure of a linear-chain CRF, like the one used here for sentiment labeling.

The random variables X and Y are jointly distributed, but in the discriminative framework of CRF, a conditional model is constructed $p(Y|X)$ from paired observation and label sequences, while the

[4] We use the open-source tool Mallet, which can be found at `http://mallet.cs.umass.edu/`.

Table 1: A segmented sentence.

positive	negative	negative
Creative is an excellent mp3 player,	although expensive,	but its supplied earphones are of inferior quality

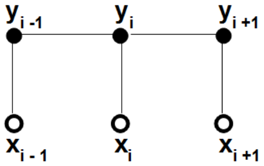

Figure 1: Example of a linear-chain CRF

marginal $p(X)$ is not explicitly modeled. The conditional probability $p(Y|X)$ is computed as

$$p(Y|X) = \frac{1}{Z(X)} \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)) \qquad (1)$$

where Z(X) is a normalization factor that implicitly depends on X and the parameters λ. In order to learn the classification model, e.g. labeling word x_i in Figure 1, CRF can use information from preceding (e.g. x_{i-1}) and following words (e.g. x_{i+1}), through *feature functions* (f_k). Feature functions are functions from a pair of adjacent output labels (states), the input sequence X and the current position in the sequence to a real value. The weight λ of each feature function is learned from the training data.

In order to train the CRF, we have annotated the benchmark corpus used in our experiments, at the segment level. Each *instance* in the dataset, i.e., each product review, is a sequence of words, labeled according to the sentiment of the segment in which they belong. We used the convention that each entity reference must be inside a text segment.

Additionally, each word is mapped onto a feature vector and thus each instance is mapped onto a sequence of feature vectors. Feature vectors capture the words/tokens and their part of speech within a window around the word of interest. Our experiments have shown that using a window of three words before and after the classified word in the vector provides sufficient context information. Thus, the vector for each word consists of 14 features, corresponding to the 7 words within the running window and their parts of speech, and it is labeled with the sentiment expressed in the segment towards the entity.

As an example of the construction of the training data, let us assume the following piece of text:

> "[...] expert amateur could want. But at the same time,
> it takes wonderful pictures very easily in."

The feature vectors for the words "want", "at" and "time" will take the form shown in Table 2.

The word of interest is given by the feature "word$_i$". A question mark "?" represents the absence of a feature at that position of the vector. Neighboring words beyond the limits of a sentence are not considered, as they often generate noise. Thus the word "but" has no preceding words. Note that this does not affect the sequential nature of the classification, in that the feature vectors remain adjacent in the input sequence, i.e. there is one vector for each word.

In addition to the input sequence, instantiating variable X as explained above, the CRF is also provided with a label sequence for each *instance*. This label sequence, which instantiates variable Y of the model, results from the annotation of sentence segments. The feature vectors in Table 2 include the label sequence, in the last column. Having instantiated X and Y from a number of training documents, the CRF learns to classify word sequences into sentiment label sequences. Thus, given a new sequence of words/tokens in the feature vector representation shown in Table 2, the CRF produces the required sentiment flow of the document, i.e., the last column of Table 2.

3.3 Entity-level classification

At run-time, the CRF model generates a flow of sentiment labels for each document. Based on this flow, there are several ways to identify the sentiment expressed for a particular entity reference, i.e., an occurrence of the entity's name or an anaphora to it. In this work we adopt a simplistic approach, which is based on our definition of the segment as a homogeneous sequence of sentiment labels that contains at most a single entity reference. Assuming that we know the position of each entity reference in the sequence, the entity is assigned the sentiment label of the entity reference, e.g. of the segment in which it belongs.

In the example provided in Table 1, assuming that the CRF has produced the correct sentiment labels, the word "earphones" would be annotated as *negative* and therefore the entity "earphones of Creative mp3 player" would also be assigned negative sentiment. Regarding the identification of the entity references in the text, we assume that these are provided by a separate entity extraction process. Therefore, our dataset is pre-annotated with entity references.

Clearly the sentiment flow produced by the CRF provides rich information that can be used to identify more complex sentiment expressions about entities. This is a line of future work that we are currently pursuing. However, as shown in section 4, even the simplistic entity annotation approach adopted here provides encouraging results.

3.4 Sentiment pattern discovery

On the basis of the sentiment flow produced by CRF, one can try to identify interesting patterns of sentiment change. This is in the spirit of alternations studied in [2], but it allows the discovery of a variety of different patterns, based on statistics. In this work we have adopted an error-driven analysis approach. In particular, we looked for correlations between sequences of predictions and certain types of classification error. These patterns may provide opportunities for improving the results of sentiment analysis.

For this task, we used a variant of the Apriori algorithm to extract the most frequent patterns in the sentiment. We are only interested in the alternations of different sentiments at a sentence level, thus considering the sequences of word-level sentiment labels "pos-pos-pos-neg-neg" and "pos-neg-neg-neg-neg" both as instances of the sentiment change pattern "pos-neg". We then computed the degree of correlation between these patterns and certain types of error (e.g. a positive opinion being classified as negative) in entity-level classification. As error type "$y_t \to \hat{y}_f$", we define the case in which a reference to an entity is being misclassified as $\hat{y}_f$ while its true label is y_t. We also define the probability of this type of error as the joint probability $P(\hat{y}_f, y_t)$. To measure the degree of correlation, we use the odds ratio measure $r = \frac{P(y_t \to \hat{y}_f|Y)}{P(y_t \to \hat{y}_f)}$, i.e. the ratio of the odds of an error type $y_t \to \hat{y}_f$ occurring inside an output sequence Y,

Table 2: Example feature vectors.

word$_{i-3}$	tag$_{i-3}$	word$_{i-2}$	tag$_{i-2}$	word$_{i-1}$	tag$_{i-1}$	word$_i$	tag$_i$	word$_{i+1}$	tag$_{i+1}$	word$_{i+2}$	tag$_{i+2}$	word$_{i+3}$	tag$_{i+3}$	*label*
expert	NN	amateur	NN	could	MD	want	VB	.	.	?	?	?	?	*negative*
?	?	?	?	but	CC	at	IN	the	DT	same	JJ	time	NN	*neutral*
at	IN	the	DT	same	JJ	time	NN	it	PRP	takes	VBZ	wonderful	JJ	*neutral*

to the odds of this error type occurring in the whole dataset. The results of this analysis are presented in section 4.5.

4. EXPERIMENTAL RESULTS

In this section, we present the results of the experiments that we did with the proposed method. Section 4.1 describes how we annotated the data for training and section 4.2 how the experiments were set up. Then sections 4.3 and 4.4 present the results at the level of single-word labeling and at the level of entities. Section 4.5 presents the results of the error analysis based on sentiment pattern discovery. Finally, section 4.6 presents the results of an initial experiment that tests the dependence of the model to the domain on which it was trained.

4.1 Data annotation

For our experiments, we used the "Customer Review Data" corpus introduced in [3]. It contains 314 online reviews for 5 different products. Hu & Liu have annotated each sentence of the dataset with the sentiment expressed for the entities mentioned in the sentence, which may be either products or features of products. Thus each sentence is annotated with zero or more pairs (entity, sentiment). The "Customer Review Data" include 2108 such pairs, of which 1363 entities have positive and 745 entities negative sentiment.

Since the corpus was only annotated with entity-level sentiment labels, we had to perform the word-level sentiment annotation manually. For the annotation of the segments with sentiment labels we used the open-source Ellogon text engineering platform.[5] Ellogon provides an easy-to-use graphical tool for manual annotation of text segments, incorporating an intuitive click-and-drag method. Ellogon was also used for sentence splitting, tokenization and part-of-speech tagging of the data. Table 3 presents the distribution of sentiment labels at the word level.

Table 3: Distribution of word-level sentiment labels.

Positive	*Negative*	*Neutral*	*Total*
26418	22432	23611	72461

As mentioned above, for the purposes of our experiments, the entities which are referred in each sentence are considered known. We compared the gold-standard sentiment set by Hu & Liu with the labels assigned by our own annotation for each entity reference and found 13% disagreement. This disagreement is mainly due to entity references that were classified as neutral according to our annotation. The original dataset contains only positive and negative sentiment labels. Neutral (objective) sentences have not been labeled. Furthermore, the original data provide a single label for each entity in each sentence. However, each sentence may contain more than one entity and sometimes more than one reference to the same entity. For instance, the sentence *"The nokia 6100 is in many ways better than the 5100, but it has a smaller screen."* refers twice to *nokia 6100* and once to *nokia 5100*. The sentiment of each segment of the sentence may vary, even for the same entity. For instance, there may be a neutral statement about the entity, before a positive or a negative one, e.g. *"It is an average camera, with a great price tag"*. In this case the original data refers only once to the entity, assigning to it the sentiment that dominates the sentence.

In order to make our results directly comparable with those reported in [3], we forced 100% agreement of the entity-level annotations, by revising the classification of the segments that did not match the gold standard. In those cases, where the disagreement stemmed from the fact that there were two references with contradicting sentiments for the same entity in the same sentence, we simply chose the reference whose sentiment conformed to the gold standard. Both versions of the dataset are provided for further experimentation at the following address `http://users.iit.demokritos.gr/~paliourg/datasets/ELS.zip`.

4.2 Experimental set-up

The annotated data was used to train and test a linear-chain CRF with the Mallet toolkit. In particular, we conducted two experiments. The first one used three sentiment class labels (*positive*, *negative*, *neutral*), while the second one only two (*subjective*, *objective*). For the purposes of the second experiment, we merged the positive and negative classes of the initial labeling into *subjective* and mapped *neutral* onto *objective*. In addition to the experiments at the level of entities, we measured the performance of our method at the level of single words, using our own annotation of the data, as gold standard.

In order to obtain an as much unbiased estimate of the performance of the system as possible, we used ten-fold cross-validation. In other words, we randomly split the dataset into ten parts, each part containing one tenth of the instances. We performed ten experiments, training the model with nine tenths and testing with the remaining one tenth of the instances. All ten test sets were different and their union corresponded to the complete dataset. Thus, we obtained a label for each instance of the dataset, which we compared against the gold standard.

For the evaluation, we use the standard IR measures of recall, precision and F1-measure per class. Additionally, we used the macro-average versions of the three measures and accuracy to obtain an overall result.

Recall is computed by the formula $\frac{TP}{TP+FN}$, where TP= true positives and FN= false negatives for each class.

Precision is computed by the formula $\frac{TP}{TP+FP}$, where TP = true positives and FP = false positives for each class.

F1-measure is computed by the harmonic mean of precision and recall for each class, therefore $\frac{2*Precision*Recall}{Precision+Recall}$.

Macro-average recall, precision and F1 simply provide an unweighted average of the corresponding measures for the individual classes.

Accuracy is computed by the formula $\frac{\text{Correct classifications}}{\text{All instances}}$ over all classes.

[5] `http://www.ellogon.org/`

At the level of entities, where our ground truth is provided by the original annotation of Hu & Liu, we can only measure the performance on *positive* and *negative* label annotation, despite the fact that the model has been trained to provide also *neutral* word labels. Furthermore, for the task of classifying sentences as *subjective* or *objective*, we measure only the recall of the *subjective* class, as we have no ground truth for *objective* sentences.

4.3 Word-level classification results

First, we examine the performance of the method at the level of word labeling, using our own annotation of the data. At this level, no competing approaches against which to compare our method were available. Nevertheless, the evaluation provides insight into the performance of the proposed method and forms a basis for the interpretation of the results presented in the next section.

Table 4 presents the results for the three classes (*positive*, *negative*, *neutral*), while Table 5 for the two classes (*subjective*, *objective*). The overall results obtained for the binary classification task are higher than the three-class problem. This is mainly due to the bias of the data towards *subjective* statements, which becomes clear by the much better performance on the *subjective*, rather than the *objective* class. Given the difficulty of the three-class labeling problem at the fine grain of words, the results are also considered satisfactory. They form a good basis for coarser types of labeling, such as for entities or sentences. Furthermore, we observe a good balance between recall and precision for all three classes, despite their uneven distribution in the training data.

Table 4: Word-level 3-class sentiment classification.

	Positive	*Negative*	*Neutral*	*Total*
Recall	62.6%	52.2%	53.7%	56.2%
Precision	62.8%	52.5%	53.2%	56.2%
F1-measure	62.7%	52.3%	53.4%	56.1%
Accuracy				56.5%

Table 5: Word-level 2-class sentiment classification.

	Objective	*Subjective*	*Total*
Recall	49.3%	80.3%	64.8%
Precision	54.8%	76.6%	65.7%
F1-measure	51.9%	78.4%	65.3%
Accuracy			70.2%

4.4 Entity-level annotation results

Based on the sentiment flow generated by the word-level sentiment annotation, we decide on the sentiment expressed about entities using the simplistic method described in section 3.3. The ground truth that we use in this experiment is the original annotation of the data, comprising only two labels: *positive* and *negative*. Table 6 presents the confusion matrix of the classification. It is worth emphasizing here that our method annotates some entities as neutral, since it has been trained with such data. In particular we obtain 14.4% neutral entities. However, the gold standard does not contain neutral labels and, therefore, the corresponding results count towards the errors of our method. We have deliberately avoided modifying our method to train on a binary classification task.

Table 7 compares our results to those published previously in [3]. In order to perform this comparison, we have calculated the expected accuracy of that method, based on its recall on extracting opinion sentences (69.3%) and its accuracy in separating *subjective* sentences into *positive* and *negative* (84.2%). In practice, the method presented in [3] will first need to identify the sentences containing opinion about entities, missing 30.4% of the total opinion-carrying sentences, and then perform a binary sentiment classification of these into positive and negative classes at an accuracy of 84.2%. Thus, its overall accuracy on this set of subjective sentences will be 69.3%×84.2%=58.4%. In contrast, and despite the fact that it is not trained for binary classification, our method achieves an accuracy of 68.6%.

The performance of our method seems also competitive to more recent results on the same dataset presented in [11] and [2]. A direct comparison cannot be made with the former as the data was re-annotated for that study and the re-annotated data are not publicly available[6]. Concerning the latter, the authors only deal with opinion orientation, assuming the absence of non-opinionated sentences, which is a different task from ours.

Table 6: Confusion matrix for entity-level 3-class sentiment classification.

		Predicted		
Actual	*Positive*	*Negative*	*Neutral*	*Total*
Positive	1026	178	159	1363
Negative	178	419	148	745
Neutral	0	0	0	0
Total	1204	597	307	2108

Table 7: Entity level sentiment classification

	Our method	Hu's method
Accuracy	68.6%	58.4%
F1-measure for positive class	79.9%	
F1-measure for negative class	62.4%	

One other comparison that is possible with the method in [3] is in terms of recall of subjective sentences. For this purpose, we retrained our method by merging the *positive* and *negative* classes into a single *subjective* class. As a result, the number of entities that received an *objective* label was reduced slightly from 14.4% to 12.2%. Table 8 compares this result to the recall reported in [3], which was much lower. The recall rate of our method is also higher than that reported in [11], but as explained above a direct comparison is not possible, due to the unavailability of the data.

Table 8: Entity-level opinion extraction

	Our method	Hu's method
Recall	87.8%	69.3%

[6] Following personal communication with the authors, it was not possible to retrieve the re-annotated data.

4.5 Pattern discovery results

The results of the entity-level labeling experiment illustrate the value of the sentiment flow for this task. Given the richness of the flow, we believe that it can be used for various other sentiment analysis tasks, apart from the three-class sentiment classification examined in section 4.4. As mentioned in section 3.4, we have searched for interesting sentiment alternation patterns in the data, aiming to recover some recurring errors of the sentiment classifier.

Table 9 presents the most frequent alternation patterns together with the odds ratio of every error type co-occurring with the pattern. We chose not to include patterns of length 1 (i.e. segments of uniform sentiment), as their large number reduces the amount of information that they carry. The error types that we studied are the following: pos→neg, neg→pos, pos→neu, neg→neu, where pos stands for positive, neg for negative and neu for neutral. Errors neu→neg and neu→pos have zero probability as the training data do not contain entities initially labeled with the sentiment "neutral". Therefore, we do not present any results for these patterns. Furthermore, the absence of a probability for an error type "$y_\mathrm{t} \rightarrow \hat{y}_\mathrm{f}$" described by "-" in Table 9, indicates no co-occurrence of the sentiment label $\hat{y}_\mathrm{f}$ with the pattern in the data.

The results provide several interesting correlations of error types with patterns. For instance, the occurrence of the quite frequent pattern "neg-pos" increases the probability of the error "pos→neg". Given a reference "A" to an entity for which the classifier has predicted a label "negative", the probability that "A" should be labeled with the sentiment "positive" doubles, if "A" occurs inside a "neg-pos" pattern.

By observing the sentiment labels that are missing from each pattern, we can obtain useful insight into our classifier. In those cases, we observe that errors of the form $l \rightarrow k$, when l is missing from a pattern, are less probable than in the general case. For instance, the output pattern "neu-pos-neu-pos" decreases the probability of "neg→pos" and "neg→neu" errors (ratio 0.898 and 0.288 respectively), which can be explained by the fact that the label "neg" does not occur in the pattern. Similarly, the occurrence of the pattern "neu-neg-neu", which does not include the label "pos", increases the probability of the error "neg→neu" (ratio 1.450) and decreases the probability of the errors "pos→neg" and "pos→neu" (ratios 0.665 and 0.689 respectively). Therefore, the absence of a label l from a pattern that contains sentiment alternations adds significant confidence to the event that l was also absent from the original sequence.

4.6 Domain independence experiment

The last experiment that we conducted aimed to show that much of the information implicit in the sentiment flow is not completely domain-dependent. For this purpose, we ran four training-test experiments, where each time we trained the classifier using only the reviews belonging to three of the four product types (e.g. dvd player, mp3 player, mobile phone) and tested it on the reviews belonging to the fourth type (e.g. cameras). Reviews for different product types use a substantially different vocabulary of features and characteristics between them. The results that we obtained in this initial experiment are also very encouraging. As shown in Table 10, the average accuracy of the method on the reviews in this domain independence experiment did not drop dramatically compared to the average random-split ten-fold cross-validation accuracy for all five products (copied from Tables 4 and 7), where training and test sets were allowed to contain different reviews for the same product. This demonstrates that our method could be applied to reviews about new, unseen products with little need for re-training.

Table 10: Domain independence experiment. Column 1 presents average word-level and entity-level accuracies for the experiment, each fold containing a different product type. For comparison, column 2 presents accuracies for the original, random-split ten-fold cross-validation on the whole dataset.

	Average accuracy in domain independence experiment	*Random-split 10-fold cross-val for all products*
Word-level accuracy	53.2%	56.5%
Entity-level accuracy	61.7%	68.6%

5. CONCLUSIONS

In this paper we presented a new method for entity-level sentiment classification, based on sequential modeling with Conditional Random Fields (CRF). In contrast to state-of-the-art approaches, the proposed method classifies the sentiment of each word in the document, based on the sequence of preceding words and their own sentiment. Using the order of words in this manner, the method achieves significantly higher performance on small pieces of text, than the results reported in the literature. This is particularly important for entity-level classification, where we are usually after local sentiment, expressed in a small part of a large document, e.g. a blog post or a product comparison article. Based on an initial experiment, our method seems also reasonably independent of the domain on which it is trained.

We believe that the proposed approach to sentiment classification of word sequences has a variety of applications beyond single-entity sentiment analysis. It provides a complete sentiment flow over the document that is being analyzed. This flow can be used to identify higher-level patterns of interest, such as comparison patterns between entities and other interesting linguistic forms. We have investigated the correlation of one particular type of pattern (sentiment alternations) with different types of error made by our classifier. We are currently studying other uses of the information flow.

At the same time, we are trying to reduce the effort required for manual annotation of the training data for the CRF. For this purpose, we are studying semi-supervised and active learning approaches that can be seeded with a small set of labeled examples and extract useful statistics from a larger corpus of unlabeled documents.

Finally, we are seeking to test the method with more and larger data in new domains.

Acknowledgments

This work was partially funded by the EC, in the context of the project SYNC3 (contract number: FP7-231854).

6. REFERENCES

[1] E. Breck, Y. Choi, and C. Cardie. Identifying expressions of opinion in context. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*, Hyderabad, India, Jan. 2007.

[2] X. Ding and B. Liu. The utility of linguistic rules in opinion mining. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812, New York, NY, USA, 2007. ACM.

[3] M. Hu and B. Liu. Mining and summarizing customer reviews. *In Proceedings of ACM Special Interest Group on*

Table 9: Most frequent patterns (column 2), ordered by number of appearances (column 3) and the odds ratio of every error type occurring in the sequence of the pattern (column 4-7). Patterns from which label "pos" or "neg" is absent are highlighted.

#	Pattern	number of appearances	pos→neg	pos→neu	neg→pos	neg→neu
1	neu-pos	387	-	1.389	1.020	0.720
2	**pos-neu**	**296**	**-**	**1.489**	**0.809**	**0.671**
3	pos-neg	285	1.194	-	1.177	-
4	**neg-neu**	**239**	**0.796**	**0.784**	**-**	**1.276**
5	neu-neg	235	0.841	0.805	-	1.243
6	neg-pos	219	1.997	-	1.597	-
7	**neu-pos-neu**	**161**	**-**	**1.862**	**0.931**	**0.537**
8	**pos-neu-pos**	**153**	**-**	**2.269**	**0.708**	**0.441**
9	neu-pos-neg	142	1.126	2.281	0.788	0.438
10	neg-neu-pos	130	0.792	1.227	1.494	0.815
11	pos-neg-pos	124	1.999	-	1.393	-
12	pos-neg-neu	116	0.929	0.765	1.523	1.308
13	**neu-neg-neu**	**113**	**0.665**	**0.689**	**-**	**1.450**
14	pos-neu-neg	105	0.942	0.889	1.124	1.125
15	neg-pos-neg	102	1.634	-	2.496	-
16	**neu-pos-neu-pos**	**84**	**-**	**3.469**	**0.898**	**0.288**
17	neu-neg-pos	81	2.008	1.130	1.656	0.885
18	**neg-neu-neg**	**78**	**0.869**	**0.621**	**-**	**1.611**
19	neg-pos-neu	73	2.354	1.210	1.491	0.826
20	pos-neg-neu-pos	69	0.888	1.210	1.161	0.826
21	neu-pos-neg-pos	62	1.545	3.723	1.312	0.269
22	**pos-neu-pos-neu**	**62**	**-**	**2.194**	**0.549**	**0.456**
23	neu-pos-neg-neu	59	0.920	0.731	0.922	1.367
24	pos-neg-pos-neg	57	1.988	-	1.466	-
25	neu-pos-neu-neg	57	0.560	1.225	1.441	0.816
26	pos-neu-pos-neg	56	0.785	4.189	0.745	0.239
27	neu-neg-neu-pos	55	0.689	1.110	2.009	0.901
28	pos-neu-neg-neu	51	0.314	0.349	1.844	2.865

Knowlenge Discovery and Data Mining, pages 168–177, 2004.

[4] N. Jakob and I. Gurevych. Extracting Opinion Targets in a Single-and Cross-Domain Setting with Conditional Random Fields. *In Proceedings of Empirical Methods in Natural Language Processing*, pages 1035–1045, 2010.

[5] J. Lafferty, F. Pereira, and A. McCallum. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In International Conference on Machine Learning*, 2001.

[6] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[7] Y. Mao and G. Lebanon. Isotonic conditional random fields and local sentiment flow. *In Advances in Neural Information Processing Systems*, 2007.

[8] T. Nakagawa, K. Inui, and S. Kurohashi. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794. Association for Computational Linguistics, 2010.

[9] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[10] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *In Proceedings of Empirical Methods on Natural Language Processing.*, 6:79–86, 2002.

[11] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. *In Proceedings of Empirical Methods on Natural Language Processing*, pages 339–346, 2005.

[12] K. Sadamitsu, S. Sekine, and M. Yamamot. Sentiment analysis based on probabilistic models using inter-sentence information. *In Proceedings of the Sixth International Language Resources and Evaluation*, 2008.

[13] M. Sharifi and W. Cohen. Finding domain specific polar words for sentiment classification. *Presented in Language Technologies Institute Student Research Symposium*, 2008.

[14] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *In Proceedings of the Fortieth Annual Meeting of the Association for Computational Linguistics*, 2002.

[15] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 2003.

[16] J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing*, pages 486–497, 2005.

[17] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. *In Proceedings of Empirical Methods on Natural Language*

Processing, pages 347–354, 2005.

[18] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *In Proceedings of Empirical Methods on Natural Language Processing.*, 2003.

[19] J. Zhao, K. Liu, and G. Wang. Adding redundant features for CRFs-based sentence sentiment classification. *In Proceedings of Empirical Methods in Natural Language Processing*, pages 117–126, 2008.