Proceedings of the 2011 IEEE 12th International Conference on Data Mining Workshops, SENTIRE 2013. Dallas, TX, USA. IEEE Computer Society.
DOI 10.1109/ICDMW.2013.13

Joint and Pipeline Probabilistic Models for Fine-Grained Sentiment Analysis: Extracting Aspects, Subjective Phrases and their Relations

Roman Klinger and Philipp Cimiano
Semantic Computing Group
Cognitive Interaction Technology – Center of Excellence (CIT-EC)
Bielefeld University
33615 Bielefeld, Germany
Email: {rklinger,cimiano}@cit-ec.uni-bielefeld.de

***Abstract*—Sentiment analysis and opinion mining are often addressed as a text classification or entity recognition problem, involving the detection or classification of aspects and subjective phrases. Many approaches do not model the relation between aspects and subjective phrases explicitly, implicitly assuming that a subjective phrase refers to a certain aspect if they co-occur together in the same sentence, thus potentially sacrificing accuracy. Instead, in the approach presented in this paper, we model the relation between aspects and subjective phrases explicitly, exploiting a flexible model based on imperatively defined factor graphs (IDF). The extraction of subjective phrases, aspects and the relation between them is modeled as a joint inference problem and compared to a pure pipeline architecture.**

Our goal is to analyse and quantify to what extent a joint model outperforms a pipeline model in terms of extraction of aspects, subjective phrases and the relation between them. Our results show that, while we have a substantial improvement on predicting targets using a joint inference model, the performance on subjective phrase detection and relation extraction actually decreases only slightly.

***Keywords*-fine-grained sentiment analysis, probabilistic graphical models, factorie, imperatively defined factor graphs, information extraction, machine learning**

I. Introduction

Sentiment analysis or opinion mining is the task of identifying the opinion about specific entities, products or persons in text. Typical sources considered for sentiment analysis are Twitter[1], reviews from Amazon[2] or other more domain-specific discussion forums. Often, opinion analysis is approached as a text classification task in which snippets (like tweets or sentences) are retrieved by occurrence of a keyword and the surrounding text is then categorized into being objective or subjective and in the latter case positive, negative, or neutral [1]–[3]. More valuable and differentiated results can be obtained by approaches segmenting text into phrases which denote a specific aspect or an accompanying subjective expression of some polarity [4]–[10].

Towards a more fine-grained analysis of the opinions expressed, it seems crucial to capture the relation between subjective phrases and aspects they refer to explicitly. In most state-of-the-art approaches, this relation is typically only implicit, assuming that a subjective phrase refers to aspects that it co-occurs with within a sentence or some other unit. This is in general too course and might lead to low accuracy.

Consider the following example:

At least, the weight of this ugly camera is great.

The sentence mentions two aspects: weight and camera as well as three subjective expressions, two positive, *i. e.* great and At least, and one negative, *i. e.* ugly. Failing to capture the explicit relation, *i. e.* that ugly refers to camera and great to weight is clearly problematic. This motivates the need for deeper linguistic analysis, detecting that weight is a target of great, and camera is a target of ugly. At least might be considered negative regarding previous text or positive regarding the current sentence.

Only little work has been performed in extracting sentiment and opinion-related information in such a fine-grained manner. Most work in the area of opinion mining has concentrated on either predicting one of these variables in isolation (*e. g.* subjective expressions [6]) or modeling the dependencies uni-directionally in a pipeline architecture, *e. g.* predicting targets on the basis of perfect and complete knowledge about subjective terms [11]. The relational structure has been analyzed in detail with multiple types of relations, but based on the assumption that entities were given [12].

Very recent work analyzed the interdependencies between targets and aspects, showing that there is a positive impact of entities in both directions (first predicting subjective phrases and then targets and the other way round) and a loss of performance if both entities need to be predicted instead of having partly perfect knowledge [13]. Similarly, an inductive logic programming approach combining real-world predictions of targets and subjective phrases in a joint fashion has been proposed recently [14]. Here, an optimization procedure selects suitable entities based on their marginal probabilities and their relation predicted by a maximum

[1] http://www.twitter.com/
[2] http://www.amazon.com/

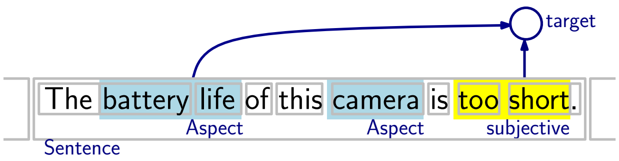

Figure 1. Data representation: spans represent the entities of type *aspect* and *subjective*. Subjective spans have an associated list of aspects they are related to. In case an aspect is described by a subjective phrase we call the latter a *target*.

entropy approach. This method could be understood as a pipeline approach in which an optimization step at the end selects best fitting combinations of the sub-solutions, which is closely related to modeling the problem as a joint inference task.

In order to yield a deeper and more fine-grained analysis of sentiments, we propose to model the relation between aspects and subjective phrases explicitly in a flexible system that relies on joint inference to predict aspects, subjective phrases and their relations. Our approach builds on factor graphs as implemented by imperatively defined factor graphs (IDF), a framework to design probabilistic graphical models with an arbitrary structure [15], [16]. Next to other inference algorithms, it includes Metropolis-Hastings sampling [17]–[19], a Markov-chain-Monte-Carlo method for inference that allows for handling huge graphical structures [20]–[22]. Thus, we gain the needed flexibility to model the interaction between different variables. We model targets, *i. e.*, aspects that are in relation with a subjective phrase, as well as subjective phrases as span variables and design our joint model to make predictions about these two variables and their relation in a single step. This design is beneficial for being augmented with other classes of interest, such as other entities or relational structures. This might include conditions under which some aspect evaluation holds as well as identification of opinion holders or other variables of interest.

We provide the following contributions in this paper:

- We present a flexible system and framework that allows to model the relation between subjects and aspects explicitly, thus allowing for a more fine-grained analysis of sentiment. The approach is flexible in that other variables, relations or substructures can be added any time.
- We study the impact of joint inference in comparison to a pipeline model, showing that for the prediction of aspects, the joint inference model outperforms the pipeline model. In the prediction of subjective phrases and relations, the pipeline model shows superior performance.

In the following, we shortly introduce imperatively defined factor graphs in Section II-A. We then explain our data structures in Section II-B and our probabilistic dependencies in Section II-C. The difference between a joint and a pipeline model is described in Section II-D. The experimental setting and results are discussed in Section III. We conclude in Section IV.

II. Methods

A. Imperatively Defined Factor Graphs

We exploit factor graphs as a method to define the probabilistic (undirected) model of the relation between input text and output variables.

A factor graph [23] is a bipartite graph over factors and variables. A factor graph G defines a probability distribution over a set of output variables $\mathbf{y}$ conditioned on input variables $\mathbf{x}$. A factor Ψ_i computes a scalar value over a subset of variables $\mathbf{x}_i$ and $\mathbf{y}_i$ that are neighbors of Ψ_i in the graph. This real-valued function can be defined as the exponential of an inner product over features $\{f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$ and parameters $\{\theta_{ik}\}$, where $k \in [1, K_i]$ and K_i is the number of parameters for factor Ψ_i, and $Z(\mathbf{x})$ is the normalization function. The probability distribution is therefore

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\Psi_i \in G} \exp\left(\sum_{k=1}^{K_i} \theta_{ik} f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\right). \quad (1)$$

Defining factor graphs statically for each data point is not convenient. In addition, generalizations can be made which limit the dimensionality. Therefore, factor graphs often share parameters between several factors. A factor template T_j consists of parameters $\{\theta_{jk}\}$, features $\{f_{jk}\}$, and a description of a relationship between variables, yielding a set of tuples $\{(\mathbf{x}_j, \mathbf{y}_j)\}$. For each of these variable tuples $(\mathbf{x}_i, \mathbf{y}_i)$ that fulfill this relationship, the factor template instantiates a factor that shares $\{\theta_{jk}\}$ and $\{f_{jk}\}$ with all other instantiations of T_j. $\mathcal{T}$ is the set of factor templates. In this case the probability distribution is

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{T_j \in \mathcal{T}} \prod_{(\mathbf{x}_i, \mathbf{y}_i) \in T_j} \exp\left(\sum_{k=1}^{K_j} \theta_{jk} f_{jk}(\mathbf{x}_i, \mathbf{y}_i)\right). \quad (2)$$

Imperatively-defined factor graphs (IDF) are an approach to probabilistic programming that preserves the declarative semantics of factor graphs, while leveraging imperative constructs (pieces of procedural programming) to support both efficiency and natural intuition in specifying model structure, inference, and learning. In particular, we exploit the FACTORIE toolkit (in version 1.0.0-M7) as an implementation of IDFs. FACTORIE relies on Markov chain Monte Carlo (MCMC) inference, a common approach for inference in very large graph structures [20]–[22]. It only requires to represent a single world at one time by proposing a change to the current world and accepting that change depending on the ratio of post- and pre-proposal model scores. Factors that involve unchanged variables do not need to be (re-)evaluated.

IDF programming consists of four stages: (1) representing data through variables, (2) designing templates that define the graphical structure of the network, (3) optionally providing application-specific hooks for efficient inference, involving the definition of operations that can be exploited by sample-based inference to evaluate different world states, (4) reading in the data, learning parameters, testing, and evaluating. These steps, for the use case of fine-grained sentiment analysis, *i. e.*, the detection of subjective terms, aspects and their relation, are described in the following sections.

B. Data Representation

Our probabilistic model consists of variables which are connected via factors, the latter being described in Section II-C. The data and variable representation is depicted in Figure 1. In our model, each token of the input text constitutes an observed variable. Each sentence is a sequence of these tokens, and the sequence of sentences form the whole document. Sentences and documents are considered variables as well. The representation of entities is similar to a semi-Markov conditional random field [24]. Aspects and subjective phrases are modeled as spans of offsets in the text sequence. In contrast to a linear chain conditional random field [25], this allows for taking distant dependencies of unobserved variables into account and simplifies the design of features measuring characteristics of multi-token phrases. The relevant variables, *i. e.*, aspect and subjective phrase, are modeled via complex span variables of the form $s = (l, r, c, \vartheta)$ with a left and right offset l and r, and a class $c \in \{\text{target}, \text{subjective}\}$. These offsets denote the span on a token sequence $\mathbf{t} = (t_1, \ldots, t_n)$. The relation is modeled as an attribute ϑ of the subjective span variables, which consists of a list of aspects which are its respective targets and is a variable itself. This attribute is always empty for aspect spans. Note that modeling the relation as part of a subjective entity is motivated by the desire to have a natural and intuitive data structure. This helps in implementation of the templates and sampling operations described later. It does not have disadvantages in comparison to storing separate relation variables as part of each sentence or document.

C. Templates

As introduced in Section II-A, templates define the sets of variables that form factors (*i. e.*, the graphical structure of the probabilistic model), the features that lead to the factor's score, and the parameters associated with them. In FACTORIE, templates are Scala classes implementing unroll methods to define the connectivity of the graphical model by returning all factors that the template associates with a specific variable.

In order to measure the characteristics of aspect and subjective spans, we incorporate three different templates. Our *single span template* defines textual features for each span in isolation. It defines factors with scores based on features of the tokens in the span and its vicinity. In our model, all features are boolean. As token-based features, we use the POS tag, the lower-case representation of the token as well as both in combination. The actual span representation consists of these features prefixed with "I" for all tokens in the span, with "B" for the token at the beginning of the span, and with "E" for the token at the end of the span. In addition, the sequence of POS tags of all tokens in the span is included as a feature.

The *inter span template* measures characteristics of each pair of an aspect and subjective span. It consists of the cross product of all features from the single span template with each of the following features (inspired by [11]). Firstly, we measure if a potential target span contains a noun which is the closest noun to a subjective expression. Secondly, we measure for each span if a span of the other class is contained in the same sentence. A third feature indicates whether there is only one edge in the dependency graph between the tokens contained in spans of a different class. For building the dependency structure, the Stanford parser is used [26].

The third template is the *relation template*. In contrast to the single span and inter span templates, it does not measure the characteristics of spans to determine their likelihood to be correct but uses their characteristics to decide whether an aspect is actually the target of a subjective phrase. The characteristics measured are the ones mentioned before in the inter span template, but without being combined with the textual features (namely, we measure if a target is the closest noun, if both entities are in the sentence and if the dependency path has an edge length of one). Therefore, only grammatical dependencies are taken into account by this template.

Note that the templates as well as variable and data description introduced here differ from the description in [13] as they make the relation extraction implicit. The relational dependency has been taken into account only to predict the target span.

We do inference by Markov chain Monte Carlo sampling, an inference procedure which is deeply integrated in FACTORIE and natural to use with the template's unroll method described above. The main difference between a pipeline architecture and a joint architecture lies on the one hand in the choice of templates/features as well as in the applied sampling procedure on the other. In the case of the pipeline architecture, only plain features that do not model the interaction between aspects and subjective phrases are used while in the joint architecture features describing the relation between those are used. In the case of the joint model, the sampling space is defined by the possible world changes. Each of these potential world changes is evaluated by means of templates and the most probable ones are selected, converging at the maximum-a-posteriori configuration.

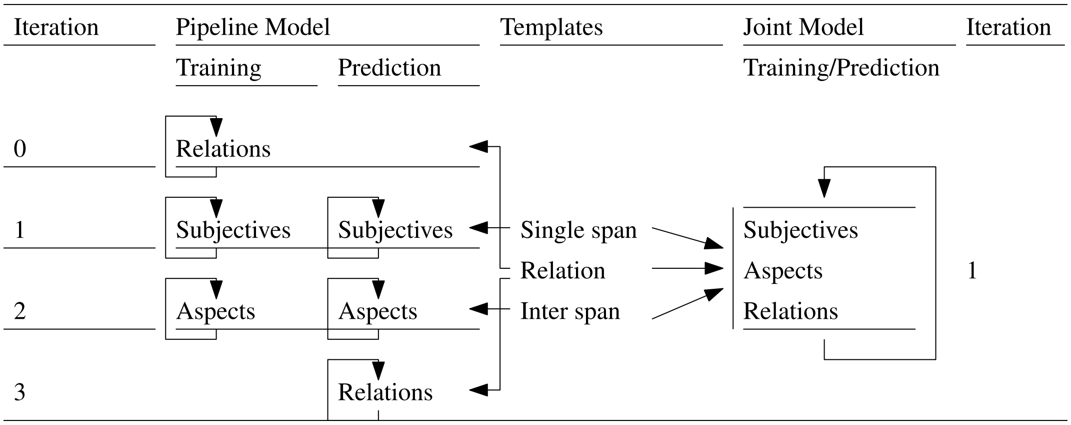

Figure 2. Three steps of the pipeline model and one step in the joint model. Each step is repeated with the number of training iteration.

D. Sampling strategies for the joint and pipeline models

In the pipeline setting, we follow a three-step procedure. The first step only deals with subjective spans. For each token, generating a span, and for each span, removing it as well as changing its length is proposed. In addition, joining a span with a succeeding or preceding span is proposed. Finally, one available state change is to do nothing. As subjective phrases are predicted first in the pipeline architecture and thus no aspect entities have been predicted up to that point, only the single span template features are taken into account in this first step.

In the second step of the pipeline, the subjective phrases are kept as they are while the proposals as described before are repeated for aspect spans. This leads to the introduction of aspects for the prediction of which the inter span template features are exploited in addition to the single span template features.

The third step is the prediction of the relations between aspects and subjective terms. In contrast to the first two steps in which the training is performed in the same manner as the actual prediction[3] (but of course not on the same data), the relation classifier is trained beforehand on spans as they are in the manually annotated data, *i. e.*, no error propagation can occur while training the relation classifier. The prediction, however, is performed on the predicted spans. For relation detection, we use a simple linear maximum entropy based classifier.

In contrast to the pipeline model, the joint model attempts to make multiple decisions at the same sampling step. This means that the set of proposals made is more complex, while only one step is performed instead of three. The proposals are intuitively the same as in the pipeline model, with the addition that proposing an aspect is done together with adding it to a subjective phrase as target. An alternative to the joint relation and entity proposal would have been to propose subjective phrases together with relations instead of proposing aspects with relations. In addition, each span operation is accompanied by correction of the relational target structure. In spite of the fact that aspects are introduced together with a relation to a subjective phrase, the introduced relations can be removed from the second iteration on. It is thus possible for aspect entities to exist without being part of a relation.

In detail, the proposals include to keep the world state as is. In addition, we propose a subjective span and an aspect span with the latter being proposed as target for each available subjective span. Thereby, when adding an aspect, the relation to subjective spans is taken into account. For each existing span on each token, the operations of removing the last token or the first token are added. For subjective spans, joining the span with a succeeding span must be accompanied by merging the targets and attaching them to the new span as well as removing them from the original spans, analogously for the preceding span. Similarly, deleting a span is accompanied by deleting the target relations, as it is when the class of the span is changed. Analogously, changing the class of an aspect span as well as deleting such a span leads to removing the aspect from all relations in which it is a target. Similarly, such operations are performed when joining with a preceding or succeeding span.

A graphical comparison of the pipeline and the joint model and their use of the templates is depicted in Figure 2.

E. Objective Functions and Training

With the definition of the sampling operations, the space to be searched for the optimal configuration is well defined. This sampling/inference strategy is also used for training. In order to learn the parameters of our model, we apply SampleRank [27]. This allows to make parameter updates within inference and hence to speed-up convergence. A crucial component in the SampleRank framework is the objective function.

In the pipeline model, the objective function for relation detection corresponds to accuracy. For span detection, we use

[3] Training is discussed in more detail in Section II-E.

the following objective function $f(\mathbf{t})$ to evaluate a proposed span $\mathbf{t}$:

$$f(\mathbf{t}) = \max_{\mathbf{g}\in\mathbf{s}} \frac{o(\mathbf{t},\mathbf{g})}{|\mathbf{g}|} - \alpha \cdot p(\mathbf{t},\mathbf{g}) \,, \tag{3}$$

where $\mathbf{s}$ is the set of all spans in the gold standard. Further, the function o calculates the overlap in terms of tokens of two spans and the function p returns the number of tokens in $\mathbf{t}$ that are not contained in $\mathbf{g}$, *i. e.*, those which are outside the overlap (both functions taking into account the class of the span). Thus, the first part of the objective function represents the fraction of correctly proposed contiguous tokens, while the second part penalizes a span for containing too many tokens that are outside the best span. Here, α is a parameter which controls the penalty.

In the joint model, we use the objective function $g(\mathbf{t})$ (see below) in order to evaluate a proposed span $\mathbf{t}$. Because of the structure of the proposals, the relation is taken into account in addition. A relation (su, ta) between a subjective phrase and a target is evaluated by

$$h(su, ta) = \max_{(su^*,ta^*)\in\mathbf{rel}^*} \begin{cases} -1 \ \text{ if } o(su, su^*) = 0 \text{ or } o(ta, ta^*) = 0 \\ \frac{1}{2}(o(su, su^*) + o(ta, ta^*)) \ \text{ else} \end{cases} \tag{4}$$

The set of all gold relations is denoted with $\mathbf{rel}^*$. Note that this function is used to evaluate the proposal of relations. To be able to prefer to not include a relation, -1 is the objective score for wrong relation proposals. The evaluation of the relations is performed jointly with the evaluation of the spans. A span $\mathbf{t}$ is evaluated by

$$g(\mathbf{t}) = \beta f(\mathbf{t}) + \sum_{(su,ta)\in\mathrm{rel}(\mathbf{t})} h(su, ta) \tag{5}$$

Here, $\mathrm{rel}(\mathbf{t})$ is a function returning the set of all relations as (su, ar) pair in which the span is a target (if the span is an aspect) or in which it participates as subjective phrase. The parameter β specifies the weight of the relation evaluation in comparison to the span evaluation itself. We empirically set it to $\beta = 0.1$, *i. e.*, we favor correctly extracted relations over correctly extracted spans in our evaluation scheme.

Table I
STATISTICS OF THE DATA SETS.

	Car	Camera
Texts	457	178
Aspects	50287	17585
Subjectives	15056	5128
Relations	13466	5005

III. EXPERIMENTS & RESULTS

A. Experimental Setting

We report results on the J.D. Power and Associates Sentiment Corpora[4], an annotated data set of blog posts in the car and in the camera domain [28]. From the rich annotation set, we use subjective terms and entity mentions as aspects. We do not consider `comitter`, `negator`, `neutralizer`, `comparison`, `opo`, or `descriptor` annotations to be subjective expressions. The relations annotated with subjective expressions are used to train our model to express which aspects take the role of targets.

A short summary of the datasets is given in Table I. The average number of subjective phrases per text is 33 in the car and 98 in the camera corpus. The average number of relations is 29.5 and 28.1, respectively. The maximal number of an entity participating in a relation is 17 for aspects referring to subjective phrases. Such extreme cases occur when positive or negative aspects are enumerated. In the case in which 7 subjective phrases refer to one single aspect, the car is described with different adjectives such as *reliable, cheap, practical, fun, large, comfortable, powerful.*

As evaluation metric, we use the F_1 measure, the harmonic mean between precision and recall. True positive spans are evaluated in a perfect match and approximate match mode, where the latter regards a span as positive if one token within it is included in a corresponding span in the gold standard. In this case, other predicted spans matching *the same* gold span do not count as false positives. The relations are evaluated similarly: A prediction is regarded as a true positive if the gold standard contains a relation in which target as well as subjective phrase is fully and exactly the same as in the prediction. In the approximate mode, a true positive *TP*(su, ta) of a subjective-phrase-target-phrase pair is 1 if and only if there exists a correct (gold) relation (su^*, ta^*) and $o(su, su^*) > 0$ and $o(ta, ta^*) > 0$.

We compare the pipeline model to the joint model on both data sets via 10-fold cross validation: We train on 8 subsets, use one for test and one for validation to select the iteration with the best result to be reported on the test set.

B. Results and Discussion

Figure 3 show the results for both the joint and the pipeline setting for cars and cameras. The darker bars correspond to perfect match, the lighter ones to the increase when taking partial matches into account. Table II shows the results in more detail. In addition, a "Relation Only" setting is presented which contains the results of a relation extractor trained and evaluated on the gold annotations of aspect and subjective spans.

The results show that it is not the case that one model outperforms the other on all subtasks (aspect recognition, subjective phrase recognition as well as relation detection).

[4] http://verbs.colorado.edu/jdpacorpus/

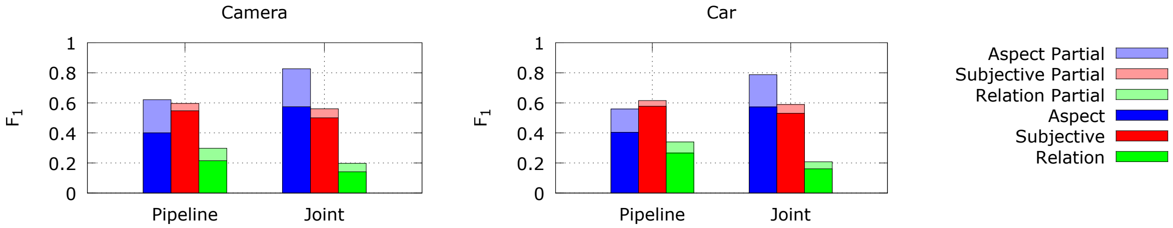

Figure 3. Results of the pipeline and the joint model for the camera data set and the car data set.

Table II
RESULTS FOR THE JOINT AND THE PIPELINE MODEL ON THE CAR AND THE CAMERA DATA SETS. NUMBERS ARE IN %.

	Exact						Partial						Exact			Partial		
	Aspect			Subjective			Aspect			Subjective			Relation			Relation		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Camera Joint	55.0	60.0	57.4	54.2	46.6	50.0	83.8	81.6	82.7	60.8	52.2	56.1	24.9	10.0	14.1	34.6	14.1	19.8
Camera Pipeline	46.7	35.1	40.0	66.0	46.7	54.7	82.3	50.0	62.1	72.1	50.8	59.6	32.0	16.2	21.5	44.3	22.5	29.8
Camera Relation Only	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	63.1	50.1	55.7	66.6	52.9	58.9
Car Joint	55.8	58.9	57.3	57.5	49.6	53.1	78.8	78.8	78.8	63.7	54.9	58.9	25.2	11.9	16.1	32.3	15.4	20.8
Car Pipeline	52.7	32.8	40.4	71.3	48.6	57.8	77.9	43.7	55.9	76.0	51.8	61.5	36.6	21.0	26.6	46.7	26.8	34.0
Car Relation Only	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	64.0	53.2	58.1	66.3	55.1	60.2

While the pipeline approach clearly outperforms the joint architecture on the extraction of relations (21.5 exact F_1 for camera, 26.6 exact F_1 for cars in contrast to 14.1 F_1 and 16.1 F_1) and slightly for subjective phrases (54.7 F_1 and 57.8 F_1 vs. 50.0 F_1 and 53.1 F_1), the joint architecture achieves higher results on the task of recognizing aspects (with 57.4 F_1 and 57.3 F_1 for camera and car over 40.0 F_1 and 40.4 F_1).

The number of entities not participating in a relation is much higher for the aspects, with 0.76 % (38464) for the car data set and 0.75 % (13145) for the camera data set in contrast to subjective phrases with 0.18 % (2703) for cars and 0.10 % (528) for cameras. This situation complicates the joint extraction, as the model needs to learn what makes an aspect not only based on the relation to the subjective phrase but in addition by means of pure textual features.

Taking joint features into account has a huge impact on aspect detection, with an increase of 16.9 percentage points for exact prediction on the car data and 17.4 percentage points on the camera data (20.6 and 22.9 percentage points for partial detection respectively) compared to the pipeline architecture. While the performance of the joint architecture is lower on the subjectivity recognition task compared to the pipeline model, the decrease in performance is relatively modest, amounting to a decrease by only 4.7 percentage points for both car data camera data (with partial match 2.6 percentage points and 3.5 percentage points, respectively). On the car data set, the difference in performance for exact relation detection is 10.5 percentage points for car data and 7.4 percentage points for the camera data.

Limiting the task to the detection of relations of perfectly known entity classes (a setting which is common in several shared tasks, *e. g.* in the BioCreative competitions for detecting protein-protein interactions [29]), the linear classifier reaches 55.7 and 58.1 F_1 for the camera and the car domain, respectively. If we take into account partially detected relations (as described above in Section III-A), we reach F-measures of 58.3% and 60.2%, respectively.

This performance might lead to the assumption that the relation extraction component of the system might be too limited to actually contribute to the whole system's performance. Another aspect is that especially the recall is limited in the setting where entities are predicted (*e. g.* the joint car model leads to 32.3 precision but only 15.4 recall). This is a typical issue for tasks which rely on previous prediction, as it is the case for relation detection on predicted entities. The observation of this characteristic in our joint model might support the assumption that the joint inference cannot fully eliminate error propagation issues in this setting.

Another aspect of the difference of the two approaches is the runtime of the system. The most important aspect of runtime is the number of proposals made in each sampling iteration, as each proposal needs two evaluations of the objective functions and of the model score to decide if the proposal is accepted or not: one to determine the current score and one for the proposed world state. The number of proposals for the joint approach is (on average) 4.92 per token, leading to 5476947 evaluations of the objective function in the whole training (with 15 training iterations). For the step of predicting subjective phrases, 2.26 proposals are made per token and 2.9 for the step of predicting aspects. This leads to 4905114 evaluations of the objective function altogether, being in the same order of magnitude.

IV. Conclusion and Future Work

We have presented a joint inference model that supports fine-grained sentiment analysis by modeling the relation between aspects and subjective phrases explicitly and modeling the statistical interaction and dependence between targets and subjective phrases. We have compared our approach to a pipeline model in which first subjective and then aspects are detected. While our joint model outperforms the pipeline model on the prediction of aspects, it performs slightly worse on the prediction of subjective phrases and clearly worse on the relation detection.

We believe the results are nevertheless interesting as they provide a novel approach to joint modeling in the context of sentiment analysis. Further, a clear benefit of our approach is that it is flexible and can be straightforwardly extended with other variables. We believe that including further subtasks might reveal the actual impact of a joint modeling approach in a clearer fashion.

The results for the joint model suggest that, in addition to the proposed model, other structures of such joint inference should be investigated. It is straight-forward to use the pipeline model, but include a joint step at the end. Similarly, pre-training of components for the joint model might have a positive impact as well. An avenue for future work would thus be to combine the advantages of both the pipeline and joint modeling approaches in one architecture.

Finally, it should be noted that the results, especially of the relation extraction can be shown to improve substantially with more informative features. We verified this by inclusion of an oracle variable which mirrors the real truth (according to the gold standard) of a prediction. This setting proves that the configuration actually is meaningful. However, future work includes the improvement of the relation detection features. In addition, understanding why the joint model performs well for certain tasks compared to the pipeline model while not for others is an important issue.

Acknowledgments

Roman Klinger has been funded by the "It's OWL" project ("Intelligent Technical Systems Ostwestfalen-Lippe", http://www.its-owl.de/), a leading-edge technology and research cluster funded by German Ministry of Education and Research (BMBF). We thank the Information Extraction and Synthesis Laboratory at the University of Massachusetts Amherst for their support and great discussions.

References

[1] O. Täckström and R. McDonald, "Semi-supervised latent variable models for sentence-level sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 569–574.

[2] A. Sayeed, J. Boyd-Graber, B. Rusk, and A. Weinberg, "Grammatical structures for word-level sentiment detection," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 667–676.

[3] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*, Barcelona, Spain, July 2004, pp. 271–278.

[4] Y. Choi, S. Kim, and S.-H. Myaeng, "Detecting Opinions and their Opinion Targets in NTCIR-8," *Proceedings of NTCIR8 Workshop Meeting*, pp. 249–254, 2010.

[5] R. Johansson and A. Moschitti, "Extracting opinion expressions and their polarities: exploration of pipelines and joint models," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers – Volume 2*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 101–106.

[6] B. Yang and C. Cardie, "Extracting opinion expressions with semi-markov conditional random fields," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 1335–1345.

[7] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 168–177.

[8] F. Li, M. Huang, and X. Zhu, "Sentiment analysis with global topics and local dependency," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, USA, 2010, pp. 1371–1376.

[9] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, October 2005, pp. 339–346.

[10] N. Jakob and I. Gurevych, "Using anaphora resolution to improve opinion target identification in movie reviews," in *Proceedings of the ACL 2010 Conference Short Papers*, ser. ACLShort '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 263–268.

[11] ——, "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*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1035–1045.

[12] Q. Zhang, Y. Wu, Y. Wu, and X. Huang, "Opinion mining with sentiment graph," in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, ser. WI-IAT '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 249–252.

[13] R. Klinger and P. Cimiano, "Bidirectional inter-dependencies of subjective expressions and targets and their value for a joint model," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013.

[14] B. Yang and C. Cardie, "Joint inference for fine-grained opinion extraction," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013.

[15] A. McCallum, K. Rohanimanesh, M. Wick, K. Schultz, and S. Singh, "FACTORIE: Efficient Probabilistic Programming via Imperative Declarations of Structure, Inference and Learning," in *Proc. of the NIPS Workshop on Probabilistic Programming*, 2008.

[16] A. McCallum, K. Schultz, and S. Singh, "FACTORIE: Probabilistic programming via imperatively defined factor graphs," in *Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 1249–1257.

[17] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[18] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[19] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[20] A. Culotta and A. McCallum, "Tractable Learning and Inference with High-Order Representations," in *Proc. of the ICML Workshop on Open Problems in Statistical Relational Learning*, 2006.

[21] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[22] B. Milch, B. Marthi, and S. Russell, "BLOG: Relational Modeling with Unknown Objects," Ph.D. dissertation, University of California, Berkeley, 2006.

[23] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Trans on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[24] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2004, pp. 1185–1192.

[25] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of the International Conference on Machine Learning*, 2001, pp. 282–289.

[26] D. Klein and C. D. Manning, "Fast exact inference with a factored model for natural language parsing," in *Proc. of Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems*, 2003.

[27] M. Wick, K. Rohanimanesh, K. Bellare, A. Culotta, and A. McCallum, "SampleRank: Training factor graphs with atomic gradients," in *Proc. of the International Conference on Machine Learning*, L. Getoor and T. Scheffer, Eds., 2011, pp. 777–784.

[28] J. S. Kessler, M. Eckert, L. Clark, and N. Nicolov, "The 2010 ICWSM JDPA Sentment Corpus for the Automotive Domain," in *Proc. od the 4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*, 2010.

[29] M. Krallinger, F. Leitner, C. Rodriguez-Penagos, and A. Valencia, "Overview of the protein-protein interaction annotation extraction task of biocreative ii." *Genome Biology*, vol. 9 Suppl 2, p. S4, 2008.