Fine-Grained Opinion Extraction with Markov Logic Networks

Luis Gerardo Mojica and **Vincent Ng**
Human Language Technology Research Institute
The University of Texas at Dallas
Richardson, TX 75083-0688
{mojica,vince}@hlt.utdallas.edu

***Abstract*—Markov Logic Networks, a joint inference framework that combines logical and probabilistic representations, enable effective modeling of the dependencies that exist between different instances of a data sample. While its ability to capture relational dependencies makes it an ideal framework for predicting the structures inherent in many natural language processing (NLP) tasks, it is arguably underused in NLP, especially in comparison to other joint inference frameworks such as integer linear programming. In this paper, we present the first Markov logic model for the NLP task of fine-grained opinion extraction that exploits a factuality lexicon. When evaluated on a standard evaluation corpus, our approach surpasses a state-of-the-art approach in performance.**

I. Introduction

Fine-grained opinion extraction is an opinion mining task that involves (1) identifying text spans corresponding to *opinions* and their *arguments* and (2) the relations between them. Compared to document-level opinion mining (e.g., determining whether a customer review is positive, negative, or neutral), fine-grained opinion extraction occurs at the sentence and phrase levels and is comparatively less investigated.

This fine-grained opinion extraction task is typically decomposed into two subtasks. The first subtask, *entity extraction*, involves identifying three types of opinionated entities, including *opinions* (expressions that explicitly reveal an internal state, such as judgment, emotion or an effective state [1]), as well as those serving as their *sources* (the entities generating the opinions) and *targets* (expressions of which the opinions are about). The second subtask, *relation extraction*, involves extracting *Is from* relations (i.e., linking a source to its opinion) and *Is about* relations (i.e., linking a target to its opinion). To understand the task, consider the following example:

> [The agency]$_{S_0}$ **considered**$_{O_0}$ that [the trade]$_{T_{0,1}}$ was favorable, but [their partners]$_{S_1}$ are **still not satisfied**$_{O_1}$

Subscripts O_0 and O_1 represent opinion spans, S_0 and S_1 indicate that the spans in brackets correspond to source entities and $T_{0,1}$ is a target. Moreover, there exists an *Is from* relation between entity S_0 and O_0 (i.e., the opinion **considered** is generated by source [The agency]), as well as between O_1 and S_1 (i.e., the opinion **still not satisfied** is generated by source [their partners]). Additionally, $T_{0,1}$ is a target entity related to O_0 and O_1 by an *Is about* relation (i.e., the two opinions, **considered** and **still not satisfied**, are both about [the trade]). Note that both opinions share the same target. In other words, it is possible to have multiple mappings between opinions and their arguments. The task is further complicated by the facts that (1) whether a word is an opinion is context-dependent (i.e. the same word can sometimes be an opinion and sometimes not); and (2) the same opinion word can be associated with more than one source/target.

A straightforward way to address this task is to make the entity extraction component first identify the entities, and then the relation extraction component determines the relation between each pair of extracted entities. However, this so-called *pipeline* approach suffers from *error propagation*, where errors made in the entity extraction component will be propagated to the relation extraction component, thus harming the performance of the latter. For example, in the example sentence above, if the entity extraction component failed to retrieve the span [the trade], it would not be possible for the relation extraction component to extract the *Is about* relations between this span and opinions O_0 and O_1.

To address this error propagation problem, Yang and Cardie [2] (henceforth Y&C) employ Integer Linear Programming (ILP) [3] to perform *joint inference* over the outputs of their entity extraction classifiers and relation extraction classifiers. Unlike in the pipeline approach, where entity extraction influences relation extraction (but not vice versa), in a joint inference approach, both tasks can influence each other. For instance, if the relation extraction component is highly confident that an *Is about* relation exists between two *candidate* entities, then these two entities will likely be extracted as an opinion and a target even if the entity extraction component fails to extract them. In other words, the final entity extraction decisions and relation extraction decisions will be made *jointly* by the two components by considering the confidence values they individually assign to the extraction decisions.

While Y&C's ILP approach has achieved the best results to date on the MPQA 2.0 corpus [1], it was evaluated in a substantially simplified setting: they removed all the sentences that do not contain any opinionated entities from both the training and test sets prior to evaluation. Hence, it is not clear how well their approach performs in practice, where many sentences do not contain any opinionated entities.

Our goal in this paper is to address fine-grained opinion extraction in a realistic setting, where we evaluate our approach *without* removing any sentence from the MPQA corpus. Unlike Y&C, we propose to employ Markov Logic Networks (MLNs) [4] for this task. MLNs are a statistical relational models that enable us to model the dependencies between different instances of a data sample. In the context of fine-grained opinion extraction, MLNs can encode the dependencies between entity extraction and relation extraction. Hence, like ILP, MLNs perform joint inference over these two subtasks.

Compared to ILP, however, MLNs are a lot less used for modeling NLP tasks. Nevertheless, MLNs have several key advantages over ILP. Not only can global constraints be specified in MLN in a more intuitive and compact manner, MLNs make it easy to specify *soft* constraints. Recall that in most existing application of ILP to NLP tasks, including Y&C's, ILP is used to enforce *hard* constraints. For instance, Y&C enforce the hard constraint that a source or target must be linked to at least one opinion expression. Now, consider the case in which the entity extractor correctly identifies a source but the relation extractor fails to link it to the corresponding opinion. Given the aforementioned hard constraint, the correctly identified source will be forced to become a non-opinionated expression. In other words, employing hard constraints does not always yield improved results: for ILP with hard constraints to improve performance, the implicit assumption is that all the underlying classifiers involved in the joint inference process are *reasonably good*. Unfortunately, employing soft constraints in ILP is not trivial.

Our goal in this paper is to employ MLNs for fine-grained opinion extraction, exploiting the ease of specifying soft constraints in an MLN. To our knowledge, this is the first MLN formulation for this task. In addition, we employ a new knowledge source for the task, the *factuality* lexicon. As we will see, this lexicon can potentially provide useful information for identifying opinion expressions that is complementary to that provided by sentiment lexicons (e.g., [5]).

Experiments on the MPQA corpus demonstrate that our MLN handily surpasses the performance of both Y&C's ILP approach, as well as a strong baseline that does not involve joint inference. Our results also suggest that fine-grained opinion extraction on the original MPQA corpus (without sentence removal) is substantially harder than the simplified MPQA that Y&C evaluated on.

II. Background

In this section, we describe related work on opinion extraction, provide details on the dataset used in our experiments and provide a gentle introduction to ILP and Markov logic.

A. Related Work

There have been many attempts to extract opinion expressions and related entities at the sentence and phrase levels. Stoyanov and Cardie [6] studied the problem of extracting entities as a summarization problem and as well as detecting coreferent entities. Wiebe et al. [1] distinguished between different types of opinion expressions, based the notion of internal states and defined targets of such opinion expressions considering their attitudes. Kim and Hovy [7] employed a semantic role labeler to detect sources of opinions and used the concept of topics as targets of opinion expressions. In a pure data driven effort, Breck et al. [8] implemented different CRFs using a variety of features with the purpose of improving opinion extraction performance. Choi et al. [9] employed a joint approach to combine the entity extraction task with relation classification, by imposing consistency constraints in the form of an ILP program. Their work inspired Y&C's ILP-based joint method, which is the state of the art in our fine-grained opinion extraction task. Ruppenhofer et al. [10] investigated the problem of extracting opinion expressions that are not necessarily explicit in a sentence and outlined new research problems in this field. Johansson and Moschitti [11] cast the opinion and source extraction tasks as a re-ranking problem. Employing a graphical method, Liu et el. [12] jointly extracted opinion expressions and targets via graph co-ranking.

B. Corpus

For training and evaluation, we use the MPQA 2.0 corpus ([1], [13]). After discarding those ill-formatted documents (lack of punctuation, paragraphs, etc.), we obtain 433 documents with $8,377$ sentences. These documents contain $4,717$ opinions, $4,680$ targets and $5,505$ sources. The number of *Is about* relations is $13,046$, and the number of *Is from* is $9,763$. Unlike Y&C, we do *not* remove sentences containing no opinionated entities.

C. Integer Linear Programming

At a high level, many NLP tasks are structured prediction problems which can be naturally expressed as constrained optimization problems, where the goal is to optimize an objective function subject to a set of linear (equality and inequality) constraints. In principle, a variety of methods can be used to solve these problems. Among them, ILP methods are a popular choice, primarily because of the following two reasons: (1) several highly optimized open source and commercial software for solving ILP problems are readily available, and therefore the application designer can focus on modeling issues rather than solving optimization problems, and (2) it is relatively straight-forward, easy and natural to express constraints in NLP as integer linear constraints. Formally, an ILP problem is defined as follows:

$$\begin{array}{ll} \text{Maximize:} & f(x_1, x_2, ..., x_n) \\ \text{Subject to:} & g_j(x_1, x_2, ..., x_n) \geq b_j \quad (j = 1, 2, ..., m) \end{array}$$

where x_i are the variables that take finite integer values, $f(x_1, x_2, ..., x_n)$ is the objective function, and g_j $(x_1, x_2, ..., x_n)$, $1 \leq j \leq m$, are the constraints (each constraint is linear in $x_1, x_2, ..., x_n$). Details of how ILP can be applied to our fine-grained opinion extraction task will be discussed in the next section.

D. Markov Logic Networks

ILP methods have a major limitation. They are propositional in nature, and are unable to model relational structure –

properties and relationships that hold across multiple objects. This makes the model specification quite cumbersome and time consuming in practice.

Markov logic ([4], [14]), a popular statistical relational learning (SRL) approach [15], remedies this problem by combining graphical models with first-order logic. At a high level, an MLN is a set of weighted first-order logic formulas (f_i, w_i). Given a set of constants that model objects in the domain, it defines a Markov network or a log-linear model [16] in which we have one node per ground first-order atom and a propositional feature corresponding to each grounding of each first-order formula. The weight of the feature is the weight of the corresponding first-order formula.

Formally, the probability of a world ω which represents an assignment of values to all ground atoms in the Markov network is given by:

$$\Pr(\omega) = \frac{1}{Z}\exp\left(\sum_i w_i N(f_i, \omega)\right)$$

where $N(f_i, \omega)$ is the number of groundings of f_i which evaluate to True in ω and Z is a normalization constant called the partition function.

The key inference tasks over MLNs are computing the partition function (Z) and the most-probable explanation given evidence (the MAP task). Most queries can be reduced to these inference tasks. Formally, the partition function and the MAP tasks are given by:

$$Z = \sum_{\omega}\exp\left(\sum_i w_i N(f_i, \omega)\right) \tag{1}$$

$$\arg\max_{\omega} P(\omega) = \arg\max_{\omega}\sum_i w_i N(f_i, \omega) \tag{2}$$

Markov logic is a popular choice for joint inference in NLP for pretty much the same reasons that ILP is popular: (1) wide availability of software packages such as Alchemy [17], Alchemy 2.0 [18], Markov the beast [19] and Tuffy [20] for inference and learning; and (2) it is easy (easier than propositional models such as ILP and graphical models which require wrapper code to either create the graphical model or the ILP problem) to compactly specify complex constraints and factors in NLP tasks using the rich first-order specification. In spite of their advantages, MLNs have been relatively underused in NLP tasks compared to ILP.

III. Baseline Systems

We employ two baseline systems.

A. Baseline 1

Our first baseline addresses the two subtasks using two independently-trained models. To train the entity extraction model, we follow Y&C, recasting the task as a sequence labeling task. Each training/test instance corresponds to a word token that is represented using the features employed by Y&C's entity extractor. These features can be broadly divided into four categories: (1) *lexical*: the token itself, its part-of-speech, and its lemma; (2) *subjectivity lexicon*: whether the token is found in the subjectivity lexicon distributed together with the MPQA corpus[1]; (3) *WordNet* [21]: the token's first hypernym, if any; and (4) *semantic*: the first FrameNet [22] frame of the token if it is a verb; null otherwise. Token features are considered in a [−4,+4] window, and the rest of the features in a [−1,+1] window. We then train a L2-regularized[2] CRF model on the training instances using CRF++[3] to identify text spans that correspond to each type of opinionated entities.

For relation extraction, we train using LIBLINEAR [23] two L2-regularized binary SVM classifiers, one for extracting *Is from* relations and the other for extracting *Is about* relations.[4] To create training/test instances, we (1) take the 30-best output from our entity extraction model[5]; (2) remove all candidate entities that overlap with those belonging to the 1-best output; (3) remove the remaining candidates one by one until there are no overlaps, giving preference to shorter candidates; and then (4) pair each candidate opinion with each candidate source/target. Each instance is represented using features employed by Y&C's relation classifiers. These features can be broadly divided into two categories: (1) *lexical*: pair of head words in the pair under consideration, pair of their part-of-speech tags; and (2) *dependency tree paths*: tokens and directions in the path between the heads of the pair of spans when traversing the dependency tree, considering only the 50 most frequent paths and discarding the rest, the dependency relations in that path using the same filtering criteria, the number of nodes traversed to reach the head of the second span, the number of candidate spans between the pair in consideration and the length of the spans.

B. Baseline 2

As our second baseline, we employ Y&C's ILP approach. As mentioned before, ILP is a constrained optimization framework, where the goal is to optimize an objective function subject to a set of linear constraints. When applied to the fine-grained opinion extraction task, Y&C combined *all* the classification decisions made by three models (namely the entity extraction model and the relation extraction classifier described in Baseline 1, as well as an *implicit* relation extraction classifier for identifying opinions with implicit arguments (i.e., opinions whose arguments are not explicitly stated in the associated text)) or each *sentence* (as well as the corresponding confidence values associated with the classification decisions)

[1]The subjectivity lexicon contains words that are manually identified as subjective.

[2]Regularization constant $c = 1000$

[3]http://taku910.github.io/crfpp/. All CRF learning parameters are set to their default values.

[4]All learning parameters are set to their default values except that we set $c = 1$ and $\epsilon = 0.0001$.

[5]CRF output on the training set is obtained via 10-fold cross validation on the training set. CRF output on the test set is obtained using the CRF trained on all training texts.

into the objective function.[6] The goal of ILP, then, is to *re-classify* the test instances associated with each sentence so that the resulting set of classifications collectively/jointly optimizes the objective function. This is a *joint* inference process in the sense that when the objective function is optimized, the test instances from *both* subtasks associated with each sentence are being re-classified *simultaneously*, rather than *independently* as in Baseline 1. It is this joint inference process that allows both subtasks to influence each other.

To get a better idea of what the objective function looks like, let us define the constrained optimization problem more formally. As mentioned above, we create one objective function for each test sentence. Specifically, for each test sentence, let O be the set of opinion candidates (obtained from the 30-best CRF output as described in Baseline 1), A_k be the set of argument candidates (also obtained from the 30-best CRF output), where k denotes the relation type (*Is about* or *Is from*), and S be the union of O and A_k.

Next, we introduce a set of binary indicator variables whose values are to be determined by ILP during the re-classification (i.e., joint inference) process. Specifically, x_{iz} has the value 1 if and only if ILP believes that span i should have entity label z; u_{ij} has the value 1 if and only if ILP believes that opinion candidate i in O has a relation with argument candidate j in A_k, and v_{ik} has the value 1 if and only if ILP believes that opinion candidate is related to a *"null"* argument of type k.

Finally, we combine these binary variables (x_{iz}, u_{ij}, and v_{ik}) with the confidence values returned by the entity extraction model and the two relation classifiers into the objective function, as shown below.

$$\arg\max_{x,u,v} \lambda \sum_{i \in S} \sum_{z} f_{iz} x_{iz} + (1-\lambda) \sum_{k} \sum_{i \in O} \left(\sum_{j \in A_k} r_{ij} u_{ij} + r_{i\emptyset} v_{ik} \right) \tag{3}$$

where the potential $r_{i\emptyset} = p(y = 1) - p(y = 0)$ is the difference in the true and false probabilities given by the implicit relation classifier regarding opinion candidate i, and the potential $r_{ij} = p(y = 1) - p(y = 0)$ is the difference in the true and false probabilities given by the (non-implicit) relation classifier over opinion candidate i and argument candidate j. As we can see, the function is a linear combination of the confidence values from the three predictors (f_{iz}, r_{ij}, $r_{i\emptyset}$), and λ is a parameter used to balance the contribution of the entity extraction component and the relation extraction component.

The objective function will be optimized subject to a set of constraints. These are constraints that we expect the re-classifications produced by ILP to satisfy. Following Y&C, we employ five constraints which can be summarized as follows: (1) each entity candidate can only be assigned exactly one of four types: opinion, target, source, or none (if it does not belong to any of the other three types); (2) among every pair of overlapping entity candidates, at most one should be extracted as an entity; (3) if an opinion candidate is predicted to be implicit, then it should not be involved in a relation with any argument candidate; if it is not implicit, it can be related to at most three sources and three targets; (4) if an argument candidate is involved in a relation, then an opinion candidate is associated with it; an argument candidate may not be related to more than three opinion candidates; and (5) if an opinion candidate is not implicit, then it must be associated with an argument candidate.[7] We solve the ILP programs using Gurobi[8]. λ is tuned to maximize F-score on development data.

IV. Factuality as a New Feature

While opinion-extraction systems, including our baselines, have extensively employed subjectivity lexicons, we propose to additionally employ a *factuality* lexicon, which we believe can provide complementary information, as described below.

Saurí [24] studied the phenomena of factuality. From her factuality lexicon we extracted 49 *categories* for 479 predicates (verbs, nouns, or adjectives) that support factual assessment about its source or target. For example, consider the two sentences "Mateo **suspects** that Luca left the country" and "Mateo **knows** that Luca the country." The two verbs, **suspects** and **knows**, belong to the categories *conjecture* and *disclose* respectively.[9] Intuitively, predicates belonging to the *disclose* category are likely to correspond to expressions involving factual instead of subjective information. On the other hand, predicates in the *conjecture* category are likely to correspond to opinions. Hence, such category information could be useful for identifying opinion expressions. To exploit such information, we train the entity extraction model with an additional *factuality* feature whose value is computed as follows. For each training/test instance, we look up the corresponding token's stem in the factuality lexicon. If found, the value of its factuality feature is the retrieved lexical category. Otherwise, the value is NA.

Note that these category labels provide a level of abstraction that enables the entity extractor to better generalize to unseen words. At the same time, they are more fine-grained than subjectivity labels and can therefore provide information not present in subjectivity labels.

V. Markov Logic for Opinion extraction

Next, we encode our MLN for fine-grained opinion extraction, OpinMLN, which is shown in Figure 1. OpinMLN contains five predicates.

Query predicates are those whose assignments are not given during inference and thus need to be predicted. We define three query predicates. `Chunk(i,l!)` is true when

[6] To train the implicit relation classifier, we follow Y&C, creating one training instance from each opinion candidate extracted from the 30-best output of the entity extraction model. Each instance is represented by a set of lexico-syntactic features encoding the opinion candidate and its surrounding context (see Y&C for details). The class value is 1 if the corresponding opinion has implicit arguments and 0 otherwise. We then train a L2-regularized binary LIBLINEAR classifier on these training instances.

[7] When used in ILP, these constraints must be encoded as linear constraints. Space limitations preclude showing these linear constraints. We refer the reader to Y&C's paper for details.

[8] http://www.gurobi.com/

[9] A predicate can be associated with multiple categories.

1) `!Is_about(i,i).`
2) `!Is_from(i,i).`
3) `!Best(i,c) v Chunk(i,c).`
4) w_4 `Is_from(i,j)` $\Rightarrow$ `Chunk(j,S)`
5) w_5 `Chunk(j,S)` $\Rightarrow$ `Is_from(i,j)`
6) w_6 `Is_about(i,j)` $\Rightarrow$ `Chunk(j,T)`
7) w_7 `Chunk(j,T)` $\Rightarrow$ `Is_about(i,j)`
8) w_8 `Overlap(i,j)` $\Rightarrow$ `(Chunk(i,N) v Chunk(j,N))`
9) w_9 `Is_from(i,j)` $\Rightarrow$ `Chunk(i,O)`
10) w_{10} `Is_about(i,j)` $\Rightarrow$ `Chunk(i,O)`

Fig. 1: The OpinMLN structure

the label assigned to text span `i` is `l`. The `!` symbol asserts that the labels assigned to a span are mutually exclusive. `Is_about(i,j)` asserts that opinion `i` is related to source `j`. Similarly, `Is_from(i,j)` asserts that opinion `i` is related to target `j`.

Evidence predicates are those whose values are known during inference. We define two evidence predicates. `Overlap(i,j)` indicates that spans `i` and `j` overlap. `Best(i,l)` is true if the label assigned to span `i` in the 1-best CRF output is `l`.

The ten MLN formulas shown in the figure express hard/soft constraints that would be desirable to enforce for this task. The first three formulas are hard formulas, meaning that they have infinite weights. We encode the remaining formulas as soft formulas. Intuitively, these are hard constraints, but as discussed before, unless the baseline models perform "reasonably" well, employing hard constraints could actually harm performance.

Formulas (1) and (2) assert that a span cannot be related to itself. Formula (3) states that the labels assigned by the 1-best CRF should not be changed by the MLN. In other words, it encodes that we are highly confident about the 1-best CRF output. Formulas (5) and (7) encode the constraint that a source/target entity must be related to an opinion. Similarly, formulas (4) and (6) encode the constraint that a related pair must consist of a source or a target, respectively. Formula (8) encodes the knowledge that given two overlapping spans, at least one is likely to be of type None (i.e., it is *not* an opinionated entity). Formulas (9) and (10) encode the constraint that the first entity in a related pair is an opinion. Note that formulas (4) and (5) can be combined into a bidirectional formula, and so are formulas (6) and (7). We encode them as separate formulas in order to allow the flexibility in assigning different weights to them.

Now that we have formulas that encode output constraints, we can incorporate the baseline models' output into the MLN. We model the entity extractor and the relation extractor's outputs as *soft evidence* in the MLN, which can be thought of as our *prior* belief that a given atom (i.e., a grounded query predicate) is true. Specifically, we include as priors the atoms `Chunk(s,l` $\neq$ `N)` with weight w_e when $p(s = l) \geq \gamma$, where $p(s = l)$ is the probability that the CRF thinks span s has entity type l. We include another atom

	Is from			*Is about*		
Experiment	P	R	F1	P	R	F1
Baseline 1	68.3	11.6	19.8	54.9	14.3	22.7
Baseline 2	33.8	9.0	14.2	24.6	16.0	19.4
B1 + factuality	73.0	13.3	**22.5**	59.9	17.2	26.7
OpinMLN + factuality	60.0	13.0	21.4	45.9	25.0	**32.4**
OpinMLN	58.8	12.8	21.0	47.7	20.3	28.5

TABLE II: Relation extraction results w.r.t. the *overlap* metric

`Chunk(s,N)` with weight w_n. In addition, we include atom `Is_from(i,j)` with weight w_r when $p(src(i,j)|x) \geq \xi$, where $p(src(i,j))$ is the probability that the relation extractor thinks opinion i and source j. In a similar fashion, we include atom `Is_about(i,j)` with weight w_r.

To ensure that the resulting weights are consistent with our intuition, we enforce the following constraints over the weight values: (1) the constraints $w_e \leq w_n$, $w_4+w_e \geq w_n$ and $w_6+w_e \geq w_n$ collectively ensure that an opinion/argument candidate receives a non-none entity label only if both the entity extractor and the (non-implicit) relation extractor say so; (2) the constraints $w_8+w_n \geq w_e+w_4+w_5$ and $w_8+w_n \geq w_e+w_6+w_7$ collectively ensure that a span that overlaps some other span should not receive a non-none entity label; and (3) the constraints $w_9 \leq w_n$ and $w_{10} \leq w_n$ ensure that in the absence of other evidences, a candidate should not receive a non-none entity label simply because the (non-implicit) relation classifier suggests that it is an opinion. Finally, w_r is tuned as follows. We (1) select an arbitrary value for w_r, (2) find the remaining weights based on the previous constraints, and (3) adjust its value to maximize F-score on development data. To achieve computational tractability, we choose a set of positive weights in the interval $[0, 1]$ via greedy search that satisfies the previous inequalities.[10]

A few points deserve mention. First, we include an atom as a prior only if its probability exceeds a certain threshold because low-confidence atoms could create noise for the learning process, thus harming performance. Second, we obtain $p(src(i,j))$ and $p(tgt(i,j))$ from the relation extractor's output directly, and compute $p(s = l)$ from the CRF output using a modified version of the forward-backward algorithm. Finally, we solve our MLN using Tuffy [20].

VI. Evaluation

For evaluation, we use the MPQA 2.0 corpus described in Section 2. Unlike Y&C, we did not correct or modify the data other than removing ill-formatted documents. This results in 433 documents, which we partition into a training set (397 documents) for model training and a test set (36 documents) for evaluation. We employ the evaluation metrics introduced by Choi et al. [9] and used by Y&C: precision, recall, and F1-score for both *overlap* and *exact* matching mechanisms[11].

[10] The resulting parameter and threshold values are: $\gamma = 0.3$, $\xi = 0.2$, w_e=0.1, w_n=0.2, w_r=0.3, w_4=0.4, w_5=0.15, w_6=0.4, w_7=0.15, w_8=0.5, w_9=0.3, w_{10}=0.35.

[11] An overlap match occurs when a predicted entity span's indices overlap with those of a gold entity.

	Overlap									Exact								
	Opinion			Target			Source			Opinion			Target			Source		
Experiment	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline 1	68.8	45.7	54.9	53.6	30.1	38.5	70.2	51.3	59.3	54.2	36.0	43.3	25.8	14.5	18.6	60.5	44.1	51.0
Baseline 2	50.5	72.2	**59.4**	44.9	36.2	40.1	67.3	37.4	48.1	39.2	56.1	46.2	15.1	12.2	13.5	57.7	32.0	41.2
B1 + factuality	71.2	47.7	57.1	56.8	28.0	37.5	74.0	53.2	61.9	57.1	38.2	45.8	26.1	12.8	17.2	62.9	45.2	52.6
OpinMLN +factuality	69.7	51.3	59.1	48.7	39.4	**43.5**	72.5	54.3	**62.1**	55.7	41.1	**47.3**	19.8	16.0	17.7	61.3	45.9	52.5
OpinMLN	75.6	45.5	56.8	55.0	34.7	42.6	77.1	49.6	60.4	59.2	35.6	44.5	25.3	16.0	**19.6**	67.7	43.6	**53.0**

TABLE I: Entity extraction results for the *overlap* and *exact* metric

Entity extraction results obtained using the overlap and exact metrics are shown in Table I. Rows 1 and 2 show the results of Baseline 1 (independently trained models for the two subtasks) and Baseline 2 (Y&C's ILP) respectively. Row 3 shows the results obtained by retraining Baseline 1 with the factuality feature. Row 4 shows the results of OpinMLN with the factuality feature incorporated and Row 5 shows OpinMLN without the factuality feature. Overall, these results are lower than those in Y&C, indicating that retaining sentences without opinionated entities yields a harder task.[12]

W.r.t. the overlap metric, ILP outperforms Baseline 1 (row 1) on Opinion and Target extraction and underperforms it on Source extraction. In other words, it is no longer the case that the use of ILP always yields improved performance. Though factuality aims to improve the extraction of Opinions, its addition to Baseline 1 not only improves Opinion extraction but also Source extraction. When used in combination with factuality, OpinMLN produces results that are better than Baseline 1 on all three types of entities, and considerably outperforms ILP on both Source and Target extraction. Finally, we can see that without factuality, the performance of OpinMLN deteriorates on all three types of entities, suggesting that factuality plays an important role in OpinMLN. Similar performance trends can be observed w.r.t. the exact metric.

Relation extraction results are shown in Table II. Following Y&C, we only report results obtained w.r.t. the overlap metric. The system configurations underlying the five rows in this table are the same as those in Table I. As we can see, Baseline 2 underperforms Baseline 1, suggesting that the use of ILP hurts relation extraction performance. Adding factuality to Baseline 1 improves the extraction of both types of relation (because factuality improves the extraction of candidate entities). Finally, compared to Baseline 1 with factuality, OpinMLN + factuality does better on extracting *Is about* relations but marginally worse on extracting *Is from* relations. Without factuality, the performance of OpinMLN deteriorates on both relation types. These results again suggest that factuality contributes positively to OpinMLN's performance.

VII. Conclusions

We proposed the first MLN formulation for the fine-grained opinion extraction task. When used in combination with factuality, our OpinMLN significantly outperforms Yang and Cardie's state-of-the-art approach on the MPQA corpus. In future work, we plan to improve OpinMLN by incorporating additional semantic knowledge, such as semantic roles.

[12]We caution that our train-test partition may not be the same as Y&C's: their partition is not available to us.

References

[1] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, 2005, pp. 165–210.
[2] B. Yang and C. Cardie, "Joint inference for fine-grained opinion extraction," in *ACL*, 2013, pp. 1640–1649.
[3] D. Roth and W.-t. Yih, "A linear programming formulation for global inference in natural language tasks," in *CoNLL* 2004, pp. 1–8.
[4] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, 2006, pp. 107–136.
[5] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *EMNLP*, 2005, pp. 347–354.
[6] V. Stoyanov and C. Cardie, "Partially supervised coreference resolution for opinion summarization through structured rule learning," in *EMNLP*, 2006, pp. 336–344.
[7] S.-M. Kim and E. Hovy, "Extracting opinions, opinion holders, and topics expressed in online news media text," in *ACL Workshop on Sentiment and Subjectivity in Text*, 2006, pp. 1–8.
[8] E. Breck, Y. Choi, and C. Cardie, "Identifying expressions of opinion in context." in *IJCAI*, vol. 7, 2007, pp. 2683–2688.
[9] Y. Choi, E. Breck, and C. Cardie, "Joint extraction of entities and relations for opinion recognition," in *EMNLP*, 2006, pp. 431–439.
[10] J. Ruppenhofer, S. Somasundaran, and J. Wiebe, "Finding the sources and targets of subjective expressions.", 2008, in *LREC*.
[11] R. Johansson and A. Moschitti, "Reranking models in fine-grained opinion analysis," in *COLING*, 2010, pp. 519–527.
[12] K. Liu, L. Xu, and J. Zhao, "Extracting opinion targets and opinion words from online reviews with graph co-ranking," in *ACL*, 2014, pp. 314–324.
[13] T. A. Wilson, "Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states," Ph.D. dissertation, University of Pittsburgh, 2008.
[14] P. Domingos and D. Lowd, "Markov logic: An interface layer for artificial intelligence," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2009, pp. 1–155.
[15] B. Taskar and L. Getoor, "Introduction to statistical relational learning," MIT press 2007.
[16] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*" MIT press, 2009.
[17] S. Kok, P. Singla, M. Richardson, P. Domingos, M. Sumner, H. Poon, D. Lowd, and J. Wang, "The Alchemy system for statistical relational AI: User manual," 2008.
[18] D. Venugopal and V. Gogate, "On lifting the Gibbs sampling algorithm," in *NIPS*, 2012, pp. 1655–1663.
[19] S. Riedel, "Cutting plane map inference for Markov logic," in *SRL*, 2009.
[20] F. Niu, C. Ré, A. Doan, and J. Shavlik, "Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS," in *VLDB*, 2011, pp. 373–384.
[21] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM* 1995, pp. 39–41.
[22] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *COLING/ACL*, 1998, pp. 86–90.
[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, pp. 1871–1874, 2008.
[24] R. Saurí, "A factuality profiler for eventualities in text," Ph.D. dissertation, Brandeis University, 2008.