SenHint: A Joint Framework for Aspect-level Sentiment Analysis by Deep Neural Networks and Linguistic Hints

Yanyan Wang, Qun Chen, Xin Liu, Murtadha Ahmed, Zhanhuai Li, Wei Pan, Hailong Liu
School of Computer Science, Northwestern Polytechnical University
Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology
Xi'an Shaanxi, P.R.China
{wangyanyan@mail.,chenbenben@,liuxin@mail.,murtadha@mail.,lizhh@,panwei1002@,liuhailong@}nwpu.edu.cn

ABSTRACT

The state-of-the-art techniques for aspect-level sentiment analysis focus on feature modeling using a variety of deep neural networks (DNN). Unfortunately, their practical performance may fall short of expectations due to semantic complexity of natural languages. Motivated by the observation that linguistic hints (e.g. explicit sentiment words and shift words) can be strong indicators of sentiment, we present a joint framework, SenHint, which integrates the output of deep neural networks and the implication of linguistic hints into a coherent reasoning model based on Markov Logic Network (MLN). In SenHint, linguistic hints are used in two ways: (1) to identify easy instances, whose sentiment can be automatically determined by machine with high accuracy; (2) to capture implicit relations between aspect polarities. We also empirically evaluate the performance of SenHint on both English and Chinese benchmark datasets. Our experimental results show that SenHint can effectively improve accuracy compared with the state-of-the-art alternatives.

KEYWORDS

Deep neural networks; Linguistic hints; Aspect-level sentiment analysis

ACM Reference Format:

Yanyan Wang, Qun Chen, Xin Liu, Murtadha Ahmed, Zhanhuai Li, Wei Pan, Hailong Liu. 2018. SenHint: A Joint Framework for Aspect-level Sentiment Analysis by Deep Neural Networks and Linguistic Hints. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3184558.3186980

1 INTRODUCTION

Aspect-level sentiment analysis, the task of extracting opinions expressed towards different aspects of an entity, is highly valuable to both consumers and businesses [7]. For example, given a review that covers two sentences *"The phone has a great resolution. But it can not last long to watch videos"* which evaluates the phone from two aspects *display* and *battery*. Our goal in this task is to identify the polarity of each aspect.

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186980

The state-of-the-art techniques focus on features modeling using a variety of deep neural networks, most popular among them is Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [10]. Despite they can achieve performance improvement compared with previous ones (e.g., lexicon-based [5] and SVM-based approaches [6]) , their practical performance may still fall short of expectations due to semantic complexity of natural languages.

Natural languages richly provide many useful linguistic hints that can effectively enhance the performance of sentiment analysis. A sentence may contain very explicit sentiment words, which are highly indicative of sentiment. In the running example, the presence of the positive sentiment word "great", together with the absence of any negative word, strongly suggest that the sentiment of the first sentence is positive. It may also contain shift words, e.g., *but* and *however*, which are highly indicative of polarity relation. Again in the running example, the word "But" at the beginning of the second sentence strongly indicates that its polarity is opposite to the polarity of the first sentence. In contrast, the absence of shift words usually suggests that the polarities of two sentences are similar.

In this demo, we present a joint framework for aspect-level sentiment analysis, SenHint, which integrates deep neural networks and linguistic hints into a coherent reasoning model. Its basic idea is to design a Markov Logic Network (MLN) for jointly modeling explicit sentiment, the output of deep neural networks, and implicit polarity relations. We note that it is *not* new to leverage linguistic hints for sentiment analysis. For instance, the classical lexicon-based approaches [5] used the hints of sentiment words for polarity reasoning; the hints of shift words have also been used to tune the performance of deep neural networks [4]. However, *SenHint* is a novel joint framework based on MLN that can integrate the output of DNN and the implication of linguistic hints in a single model. Compared with previous approaches, SenHint also uses linguistic hints in different ways. It only uses sentiment words to determine the polarities of easy instances, whose explicit sentiment can be automatically determined by machine with high accuracy. It captures implicit relations between aspect polarities by the hints of shift words and encodes them as weighted first-order logic formulas in a MLN. The major contributions of this demo can be summarized as follows:

(1) We propose SenHint, a joint framework for aspect-level sentiment analysis based on MLN;
(2) We present the techniques of encoding the output of deep neural networks and aspect polarity relations in a MLN;

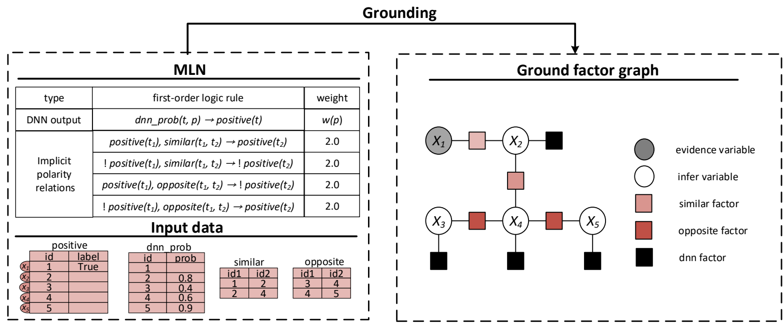

Figure 1: An example of SenHint.

(3) We empirically evaluate the performance of SenHint on both English and Chinese benchmark datasets. Our experiments show that SenHint can effectively improve accuracy compared with state-of-the-art alternatives.

The rest of this demo is organized as follows: Section 2 defines the task and presents the SenHint framework. Section 3 presents the empirical evaluation results and details demo plan.

2 SENHINT FRAMEWORK

In aspect-level sentiment analysis, the aspect, also called aspect category, is specified by a pair of entity type and attribute label. Take the phone review above as an example, there are two aspect categories in the sentences: *(display, quality)* and *(battery, operation performance)*, where *display* and *battery* are entity types, *quality* and *operation performance* are attribute labels. Following the literature [10], we view entity type as aspect. Our goal is to predict sentiment polarity of the aspects given a sentence and its pre-specified aspects.

2.1 Framework Overview

SenHint seamlessly integrates the output of DNN and the implication of linguistic hints into a unified reasoning model based on Markov Logic Network (MLN) [2]. MLN has been widely applied to infer uncertain knowledge, e.g., Deepdive [8] and ProbKB [1]. A MLN L is a set of pairs $\{(F_i, w_i)\}$, where F_i is a rule in first-order logic rule and w_i is a real number that expresses some level of confidence on this rule. Inference on MLN consists of two steps: grounding and marginal inference. The grounding process constructs a *ground factor graph* based on the rules, which defines a probability distribution over random variables. SenHint uses the marginal probability of a variable for identifying the polarity of a aspect. This process is called marginal inference of probabilistic graphical models.

An example of SenHint, including first-order logic rules and ground factor graph, has been shown in Figure 1. In the graph, aspect polarities are represented by variables (round nodes in the figure), and the influences of DNN output and linguistic implication are represented by factors (box nodes in the figure). The value of a variable indicates its corresponding aspect sentiment. There are two types of variables: *evidence variables* and *infer variables*. Evidence variables correspond to the easy aspect polarities that have been determined by explicit linguistic hints. They participate in the MLN inference, but their values are specified beforehand and remain unchanged in the inference process. The values of infer variables should instead be inferred based on the constructed MLN. Additionally, there are three types of factors: *DNN factor*, *similar factor* and *opposite factor*. The DNN factor simulates the effect of DNN output on aspect polarity. The similar factor and opposite factor represent implicit relations between aspect polarities.

Note that in SenHint, the values of evidence variables (easy aspect polarities) and implicit relations between aspect polarities are estimated based on extracted linguistic hints. In the following subsections, we will describe: 1) how to represent the output of DNN by weighted first-order rule (Subsection 2.2); 2) how to capture explicit aspect polarities using the hints of sentiment lexicon; (Subsection 2.3); 3) how to extract implicit relations between aspect polarities and represent them by weighted first-order logic rules; (Subsection 2.4); 4) how to perform joint inference on MLN (Subsection 2.5).

2.2 Modeling DNN Output

For aspect-level sentiment analysis, the approach of ATAE-LSTM [10] have attracted much attention due to its empirically improved performance compared with alternative DNNs [3]. ATAE-LSTM uses attention mechanism to concentrate on different parts of a sentence when different aspects are taken as input. Its output can indicate the influence resulting from multiple levels of features. In this demo, we train a ATAE-LSTM model and then encode its results into MLN. However, it can be observed that other DNNs can be similarly incorporated into MLN.

In order to integrate the influence of DNN output into a MLN, we design a rule, expressed by

$$w(p): \ dnn\_prob(t,p) \rightarrow positive(t) \tag{1}$$

, in which the left-hand side (LHS), $dnn\_prob(t,p)$, predicates that the probability of aspect polarity t being positive is equal to the value of p, and the right-hand side (RHS), $positive(t)$, is a boolean variable indicating whether an aspect polarity t is positive. The weight function $w(p)$ denotes the level of confidence on the rule. Observing that the relationship between the weight w and the probability p (for a boolean variable x being true) can be expressed by $p(x=1) = e^w/(1+e^w)$, we define the rule weight as

$$w(p) = ln(\frac{p}{1-p}). \tag{2}$$

According to Eq. 2, $w(p) > 0$ if $0.5 < p < 1.0$; otherwise, if $0 < p < 0.5$, $w(p) < 0$. Note that $w(p)$ takes the value of 10 when $p = 1.0$, and the value of -10 when $p = 0$. In the case of $w(p) > 0$, a zero value of $positive(t)$ would invoke a cost penalty. In the case of $w(p) < 0$, a positive value for $positive(t)$ would instead invoke a cost penalty.

2.3 Identifying Explicit Polarities

The existing lexicon-based approaches typically use the sum of sentiment values of all sentiment words in the sentence to reason about polarity. In a sentiment lexicon, the score of a sentiment word indicates its intensity of sentiment. For instance, in the lexicon of [5], the scores of sentiment words are normalized into the interval of [-4,4]: a positive (negative) score indicates positive (negative) polarity and the polarity intensity increases with the absolute value of score.

Unfortunately, the lexicon-based approaches usually fail to detect the true sentiment of a sentence in two ambiguous cases: 1) the sentence does not contain explicit sentiment words; 2) the sentiment words in the sentence hold conflicting polarities. SenHint considers a sentiment word as *explicit* if and only if the absolute value of its score exceeds a pre-specified threshold (e.g., 1.0 in our experiment). Note that an explicit sentiment word can be positive or negative. SenHint considers an aspect polarity as *easy instance* if and only if the sentence containing the aspect polarity satisfies the following two conditions:

- The sentence does not contain sentiment words with conflicting polarities;
- The sentence contains explicit sentiment words;

According to the above two conditions, the sentence, “The phone is great.”, is an easy instance, because it contains only positive words and it contains the explicit word “great”. In contrast, the sentence, “To be honest, i am a little disappointed and considering returning it”, is *not* an easy intance, because it contains both the positive word “honest” and the negative word “disappointed”.

Since negation words can effectively reverse sentiment word polarity, SenHint takes negation into consideration while reasoning about word sentiment. For each sentiment word, SenHint checks whether there is any negation word in the three words before it. If a negation word is found, word sentiment would be reversed. As a conclusion of this subsection, if an aspect polarity in a sentence is considered as easy instance, its polarity is determined by the sentiment of the explicit word in the sentence. It is modeled as evidence variables in MLN. Its polarity result would *not* be affected by MLN inference, but it could help MLN to detect the true polarity of more challenging instances.

2.4 Modeling Implicit Polarity Relations

It is common in natural languages to connect two sentences with opposite polarities by shift words. In contrast, the absence of shift words usually indicates that two neighboring sentences have similar polarities. In practice, shift words can appear within a sentence (inner-sentence) or between two sentences (inter-sentence).

In the inner-sentence case, if a sentence contains multiple aspects and no shift word, the aspect polarities within the sentence are supposed to have the same polarity. However, SenHint does not use shift words to capture opposite polarity relations, because it is very challenging to determine whether a shift word is used to connect two aspect polarities or it is only a connecting word within a phrase describing an aspect polarity. For example, the sentence "I’m not sure if it was covered in the description, but it does have a backlit keyboard and I’m getting about an 8 hour battery life on one charge" contains the shift word of “but”, but the two aspect polarities (e.g., *keyboard* and *battery*) are similar.

In the inter-sentence case, the reasoning process of SenHint works as follows: 1) if two neighboring sentences do not contain any shift word, their respective aspect polarities are supposed to be similar; 2) if neither of them contains any inner-sentence shift word but they are connected by a shift word, their respective aspect polarities are supposed to be opposite.

In SenHint, the influence of *similar* relations on aspect polarities is represented by the following two rules:

$$w_s: \ positive(t_1), similar(t_1, t_2) \rightarrow positive(t_2) \tag{3}$$

, and

$$w_s: !positive(t_1), similar(t_1, t_2) \rightarrow !positive(t_2) \tag{4}$$

, in which w_s denotes a positive weight of rule ($w_s = 2$ in our experiments), and $!positive(t_1)$ denotes the negation of a boolean variable. Similarly, the influence of *opposite* relations on aspect polarities is represented by

$$w_o: \ positive(t_1), opposite(t_1, t_2) \rightarrow !positive(t_2) \tag{5}$$

, and

$$w_o: !positive(t_1), opposite(t_1, t_2) \rightarrow positive(t_2) \tag{6}$$

, in which w_o denotes a positive weight ($w_o = w_s = 2$ in our experiments). Note that in MLN, positive rule weight means that if LHS is true, RHS tends to be also true; otherwise, rule is violated with a cost penalty.

2.5 Joint Inference

SenHint considers the output of DNN as a feature of an inference variable and transforms their relationship into a unary factor in the grounding phase. It also transforms the implicit relations defined in Eq. 3, 4, 5 and 6 into binary factors between variables. In principle, each rule should be transformed into a type of factors in the factor graph. However, the two types of factors corresponding to the rules specifying similar relation (or opposite relation) are similar; for the sake of presentation simplicity, we only show one factor between two aspect polarities in this demo. The ground factor graph then defines a probability distribution over its variables [1]. As typical in MLN inference, SenHint uses Gibbs sampling[11] for inference. In our experiments, we use the inference engine of DeepDive [8]. More details on grounding and inference are omitted here due to space limit, but can be found in our technical report [9].

3 EXPERIMENTS AND DEMO PLAN

We empirically evaluate the performance of SenHint on the benchmark datasets of SemEval 2016 Task 5 [7], including the domains laptop (English), phone (Chinese) and camera (Chinese). Our experiments perform 2-class classification to label an aspect polarity as *positive* or *negative*. Note that the laptop dataset contains neutral instances. But they are ignored in our experiments. The sentiment

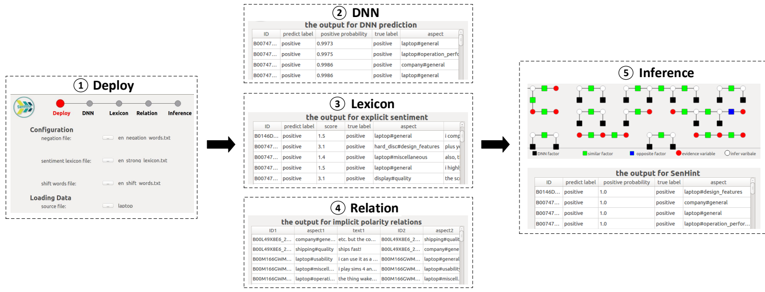

Figure 2: Screenshot for SenHint.

lexicons for English and Chinese datasets are provided by VADER [1] and the website [2] respectively. For English data, the word vectors are initialized by Glove [3]. For Chinese data, we use jieba [4] to tokenize sentences and obtain word embeddings from Baidu [5].

Table 1: Evaluation results for ATAE-LSTM and SenHint.

domain	SemEval 2016 (top)	ATAE-LSTM	Our approach		
			SH-rel	SH-lex	SenHint
laptop	—	77.35%	80%	79.20%	**81.46%**
phone	73.35%	73.91%	76.37%	77.88%	**79.58%**
camera	80.46%	80.87%	82.33%	86.07%	**87.11%**

we compare SenHint with the top performer reported on the SemEval 2016 benchmark datasets [7] as well as the state-of-the-art approach of ATAE-LSTM. To help readers to better understand the influences of respective components in SenHint, we also report the results for applying only implicit polarity relations (SH-rel), and the results for applying only explicit sentiment (SH-lex). Note that the statistics on the identified explicit sentiment and extracted polarity relations are omitted here due to space limit, but can be found in our technical report [9]. The detailed evaluation results are presented in Table 1. It can be observed that: 1) our implementation of ATAE-LSTM achieves performance very similar to the reported top performance at [7]; 2) both explicit sentiment and implicit polarity relations can effectively improve accuracy; 3) *SenHint outperforms the existing alternatives by more than 4% on* laptop *and by 5%-6% on* phone *and* camera*, which are considerable if we consider the widely recognized challenges of sentiment analysis.* Note that the results for the 2-class classification on the domain laptop is not provided in SemEval 2016, so we ignore it in our comparision.

Demo Plan. We have implemented a propotype system, whose GUI is sketched in Figure 2, it consists of five panels:

- *Deploy.* It initializes the system settings. Users can load lexicon, negation words and shift words;
- *DNN.* It uses a pre-trained DNN to predict the probability distribution for a test dataset;
- *Lexicon.* It demonstrates the results of identified explicit sentiment;
- *Relation.* It demonstrates the results of implicit polarity relations between aspects;
- *Inference.* It demonstrates the grounded factor graph of MLN and the final polarity analysis results.

The demo will be run on a laptop. The attendees will be invited to perform the polarity analysis task using different test datasets. They will be able to zoom in the results at each processing step. We have recorded a video for the demo, which can be accessed via the link: https://www.youtube.com/watch?v=3FN6GOVIBlc.

ACKNOWLEDGMENTS

This work is supported by the Ministry of Science and Technology of China, National Key Research and Development Program (2016YFB1000703), National Natural Science Foundation of China (61332006, 61732014, 61672432, 61472321 and 61502390).

REFERENCES

[1] Yang Chen and Daisy Zhe Wang. 2014. Knowledge expansion over probabilistic knowledge bases. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 649–660.

[2] Pedro Domingos and Daniel Lowd. 2009. Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3, 1 (2009), 1–155.

[3] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[4] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318* (2016).

[5] Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media.*

[6] Tony Mullen and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources.. In *EMNLP*, Vol. 4. 412–418.

[7] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, and Orphee De Clercq. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *International Workshop on Semantic Evaluation*. 19–30.

[8] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. 2015. Incremental knowledge base construction using deepdive. *Proceedings of the VLDB Endowment* 8, 11 (2015), 1310–1321.

[9] Yanyan Wang, Qun Chen, Xin Liu, Murtadha Ahmed, Zhanhuai Li, Wei Pan, and Hailong Liu. 2018. *A Joint Framework for Aspect-level Sentiment Analysis by Deep Neutral Networks and Linguistic Hints(technical report).* Technical Report. http://www.wowbigdata.com.cn/SenHint/senhint-report.pdf

[10] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Conference on Empirical Methods in Natural Language Processing*. 606–615.

[11] Ce Zhang. 2013. Towards high-throughput gibbs sampling at scale:a study across storage managers. In *ACM SIGMOD International Conference on Management of Data*. 397–408.

[1]https://github.com/cjhutto/vaderSentiment
[2]http://ir.dlut.edu.cn/EmotionOntologyDownload
[3]https://nlp.stanford.edu/projects/glove/
[4]https://github.com/fxsjy/jieba
[5]http://pan.baidu.com/s/1jIb3yr8