Knowledge-Based Systems 108 (2016) 42–49

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Aspect extraction for opinion mining with a deep convolutional neural network

Soujanya Poria [a], Erik Cambria [b,*], Alexander Gelbukh [c]

[a] Temasek Laboratories, Nanyang Technological University, Singapore
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[c] CIC, Instituto Politécnico Nacional, Mexico

ARTICLE INFO

Article history:
Received 18 November 2015
Revised 2 May 2016
Accepted 8 June 2016
Available online 17 June 2016

Keywords:
Sentiment analysis
Aspect extraction
Opinion mining
CNN
RBM
DNN

ABSTRACT

In this paper, we present the first deep learning approach to aspect extraction in opinion mining. Aspect extraction is a subtask of sentiment analysis that consists in identifying opinion targets in opinionated text, i.e., in detecting the specific aspects of a product or service the opinion holder is either praising or complaining about. We used a 7-layer deep convolutional neural network to tag each word in opinionated sentences as either aspect or non-aspect word. We also developed a set of linguistic patterns for the same purpose and combined them with the neural network. The resulting ensemble classifier, coupled with a word-embedding model for sentiment analysis, allowed our approach to obtain significantly better accuracy than state-of-the-art methods.

1. Introduction

The opportunity to capture the opinion of the general public about social events, political movements, company strategies, marketing campaigns, and product preferences has raised increasing interest of both the scientific community (because of the exciting open challenges) and the business world (because of the remarkable benefits for marketing and financial market prediction). Today, sentiment analysis research has its applications in several different scenarios. There are a good number of companies, both large- and small-scale, that focus on the analysis of opinions and sentiments as part of their mission [1].

Opinion mining techniques can be used for the creation and automated upkeep of review and opinion aggregation websites, in which opinions are continuously gathered from the Web and not restricted to just product reviews, but also to broader topics such as political issues and brand perception. Sentiment analysis also has a great potential as a sub-component technology for other systems. It can enhance the capabilities of customer relationship management and recommendation systems; for example, allowing users to find out which features customers are particularly interested in or to exclude items that have received overtly negative feedback from recommendation lists. Similarly, it can be used in social communication for troll filtering and to enhance anti-spam systems. Business intelligence is also one of the main factors behind corporate interest in the field of sentiment analysis [2].

In opinion mining, different levels of analysis granularity have been proposed, each one having its own advantages and drawbacks [3]. Aspect-based opinion mining [4,5] focuses on the relations between aspects and document polarity. An aspect, also known as an opinion target, is a concept in which the opinion is expressed in the given document. For example, in the sentence, “The screen of my phone is really nice and its resolution is superb” for a phone review contains positive polarity, i.e., the author likes the phone. However, more specifically, the positive opinion is about its *screen* and *resolution*; these concepts are thus called opinion targets, or aspects, of this opinion. The task of identifying the aspects in a given opinionated text is called aspect extraction.

There are two types of aspects defined in aspect-based opinion mining: explicit aspects and implicit aspects. Explicit aspects are words in the opinionated document that explicitly denote the opinion target. For instance, in the above example, the opinion targets *screen* and *resolution* are explicitly mentioned in the text. In contrast, an implicit aspect is a concept that represents the opinion target of an opinionated document but which is not specified explicitly in the text. One can infer that the sentence, “This camera is sleek and very affordable” implicitly contains a positive opinion of the aspects *appearance* and *price* of the entity *camera*. These same aspects would be explicit in an equivalent sentence: “The appearance of this camera is sleek and its price is very affordable.”

* Corresponding author. Tel. +65 6790 4328.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

http://dx.doi.org/10.1016/j.knosys.2016.06.009

Most of the previous works in aspect term extraction have either used conditional random fields (CRFs) [6,7] or linguistic patterns [4,8]. Both of these approaches have their own limitations: CRF is a linear model, so it needs a large number of features to work well; linguistic patterns need to be crafted by hand, and they crucially depend on the grammatical accuracy of the sentences.

In this paper, we overcome both limitations by using a convolutional neural network (CNN), a non-linear supervised classifier that can more easily fit the data. Previously, [9] used such a network to solve a range of tasks (not for aspect extraction), on which it outperformed other state-of-the-art NLP methods. In addition, we use linguistic patterns to further improve the performance of the method, though in this case the above-mentioned issues inherent in linguistic patterns affect the framework.

This paper is the first one to introduce the application of a deep learning approach to the task of aspect extraction. Our experimental results show that a deep CNN is more efficient for aspect extraction than existing approaches. We also introduced specific linguistic patterns and combined a linguistic pattern approach with a deep learning approach for the aspect extraction task.

2. Related work

Aspect extraction from opinions was first studied by Hu and Liu [4]. They introduced the distinction between explicit and implicit aspects. However, the authors only dealt with explicit aspects and used a set of rules based on statistical observations. Hu and Liu's method was later improved by Popescu and Etzioni [10] and by Blair-Goldensohn et al. [11]. Popescu and Etzioni [10] assumed the product class is known in advance. Their algorithm detects whether a noun or noun phrase is a product feature by computing the point-wise mutual information between the noun phrase and the product class.

Scaffidi et al. [12] presented a method that uses language model to identify product features. They assumed that product features are more frequent in product reviews than in a general natural language text. However, their method seems to have low precision since retrieved aspects are affected by noise. Some methods treated the aspect term extraction as sequence labeling and used CRF for that. Such methods have performed very well on the datasets even in cross-domain experiments [6,7].

Topic modeling has been widely used as a basis to perform extraction and grouping of aspects [13,14]. Two models were considered: pLSA [15] and LDA [16]. Both models introduce a latent variable "topic" between the observable variables "document" and "word" to analyze the semantic topic distribution of documents. In topic models, each document is represented as a random mixture over latent topics, where each topic is characterized by a distribution over words. Such methods have been gaining popularity in social media analysis like emerging political topic detection in Twitter [17]. The LDA model defines a Dirichlet probabilistic generative process for document-topic distribution; in each document, a latent aspect is chosen according to a multinomial distribution, controlled by a Dirichlet prior α. Then, given an aspect, a word is extracted according to another multinomial distribution, controlled by another Dirichlet prior β. Among existing works employing these models are the extraction of global aspects (such as the brand of a product) and local aspects (such as the property of a product [18]), the extraction of key phrases [19], the rating of multi-aspects [20], and the summarization of aspects and sentiments [21]. [22] employed the maximum entropy method to train a switch variable based on POS tags of words and used it to separate aspect and sentiment words.

Mcauliffe and Blei [23] added user feedback to LDA as a response-variable related to each document. Lu and Zhai [24] proposed a semi-supervised model. DF-LDA [25] also represents a semi-supervised model, which allows the user to set must-link and cannot-link constraints. A must-link constraint means that two terms must be in the same topic, while a cannot-link constraint means that two terms cannot be in the same topic. Poria et al. [26] integrated common-sense computing [27] in the calculation of word distributions in the LDA algorithm, thus enabling the shift from syntax to semantics in aspect-based sentiment analysis.

Wang et al. [28] proposed two semi-supervised models for product aspect extraction based on the use of seeding aspects. In the category of supervised methods, [29] employed seed words to guide topic models to learn topics of specific interest to a user, while [20] and [30] employed seeding words to extract related product aspects from product reviews.

On the other hand, recent approaches using deep CNNs [9,31] showed significant performance improvement over the state-of-the-art methods on a range of natural language processing (NLP) tasks. Collobert et al. [9] fed word embeddings into a CNN to solve standard NLP problems such as named entity recognition (NER), part-of-speech (POS) tagging and semantic role labeling.

3. Some background on deep CNN

A deep neural network (DNN) can be viewed as a composite of simple, unsupervised models such as restricted Boltzmann machines (RBMs), where each RBM's hidden layer serves as the visible layer for the next RBM. An RBM is a bipartite graph comprising of two layers of neurons: a visible and a hidden layer; connections between neurons in the same layer are not allowed.

To train such a multi-layer system, one needs to compute the gradient of the total energy function E with respect to weights in all the layers. To learn these weights and maximize the global energy function, the approximate maximum likelihood contrastive divergence approach can be used. This method employs each training sample to initialize the visible layer. Next, it uses the Gibbs sampling algorithm to update the hidden layer and then reconstruct the visible layer consecutively, until convergence occurs [32]. As an example, consider a logistic regression model to learn the binary hidden neurons. Each visible neuron is assumed to be a sample from a normal distribution [33]. The continuous state $\hat{h}_j$ of the hidden neuron j, with bias b_j, is a weighted sum over all continuous visible neurons $\boldsymbol{v}$:

$$\hat{h}_j = b_j + \sum_i v_i w_{ij}, \tag{1}$$

where w_{ij} is the weight of connection from the visible neuron v_i to the hidden neuron j. The binary state h_j of the hidden neuron can be defined by a sigmoid activation function:

$$h_j = \frac{1}{1 + e^{-\hat{h}_j}}. \tag{2}$$

Similarly, at the next iteration, the continuous state of each visible neuron v_i is reconstructed. Here, we determine the state of the visible neuron i, with bias c_i, as a random sample from the normal distribution where the mean is a weighted sum over all binary hidden neurons:

$$v_i = c_i + \sum_j h_i w_{ij}, \tag{3}$$

where w_{ij} is the weight of connection from the visible neuron i to the hidden one j. This continuous state is a random sample from a normal distribution $\mathcal{N}(v_i, \sigma)$, where σ is the variance of all visible neurons. Unlike hidden neurons, in a Gaussian RBM the visible ones can take continuous values.

Then, the weights are updated as the difference between the original data $\boldsymbol{v}_{data}$ and reconstructed visible layer $\boldsymbol{v}_{recon}$:

$$\triangle w_{ij} = \alpha(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}), \tag{4}$$