Using subject-predicate-object triplets for opinion mining

Daniil Feldman
MIPT
daniil.feldman@phystech.edu

Abstract

Opinion mining is a popular task, that is applied, for example, to determine news polarisation and identify product review classes. The goal is to extract opinions in a text corpus without supervision. Many algorithms that tackle this issue are based on topic models. Our goal is to find a way to increase their performance using syntactic and semantic features instead of term frequencies. More specifically, we test the hypothesis that an opinion is determined by the facts referred in it and the subjects most mentioned by the speaker. In this paper we formalise this task and prove that using SPO triplets in topic models can increase opinion mining quality.

Keywords: *Opinion mining, topic modelling, SPO triplets*

1 Introduction

Every important political event is vastly covered in the news. Most sources provide polarised texts expressing the opinion of one of the sides. As a result the people reading the news only get to know one side of the problem. We would like to provide them with all opinions on the subject and assign one of them to each news. Often there are even more then two opinions and a more general approach suggests finding the number of opinions as well. We, however, will be solving a simpler problem where this variable is given.

Topic modelling is an unsupervised technique that can find topics in a corpus of documents. Each topic is defined as a probability distribution over terms where the ones most frequently used in the topic are considered topic terms and have the highest probability (terms are the elementary objects in text, for example, words). We will determine opinions similarly, as probability distribution. That way to mine opinions we can apply topic modelling techniques. We will be working with a corpus of news covering a given event. If we train a topic model on these texts the extracted topics can be interpreted as opinions. To clarify, under opinions we will be understanding term distributions extracted by a topic model on a corpus of text covering a given event.

Using topic modelling in this way is not new. Most studies, however, use words as terms and extract dependencies on a word-level. We assume that syntactic and semantic patterns should be taken into account. Let us consider a simple example of a text consisting of just one sentence: *The new law was passed in the congress.* Each word appears in the sentence just one time, so we consider them all equally important when estimating the distributions by word frequencies. Actually, the text is about the new law so the word *law* is more important than others. The subject-predicate-object triplets in this text are: *law-was-passed, law-is-new.* The word *law* is mentioned as a subject twice, while others are mentioned only as objects. Our proposal in this paper is to count how frequent is every word mentioned as a subject and as an object.

2 Related work

Opinion mining has been vastly studied in recent years. A general survey of methods is presented in [2]. Earlier works ([1],[3]) focused on mining opinions in product reviews, but now the focus is moving towards

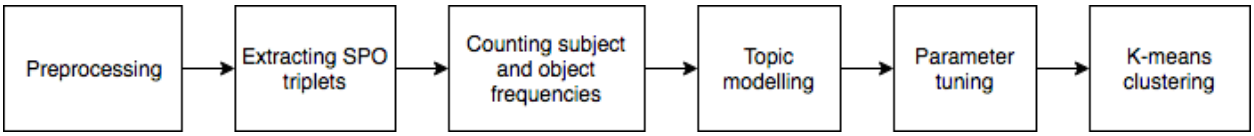
texts on political events. The discussed works rely on probabilistic topic models which are described in [2]. Topic models were used both in supervised [4] and unsupervised approaches [5]. Some works researched a more general problem of finding topics and opinions at the same time. The authors of [6] propose a topic-aspect model that mines topics and aspects, where the latter can be interpreted as opinions. None of those works studied opinion mining in Russian texts, but we will be applying similar techniques.

Study [7] was aimed at solving a different task: ontology mining. It's authors, however, used similar topic modelling techniques. They proved that using SPO triplets to build hierarchical topic models provides a gain in model quality. That gave us the hypothesis that a similar trick could be used in opinion mining.

To build topic models we will be using additive topic modelling regularizers (ARTM), a technique that allows to configure different topic models by adding regularizers. It is observed in [8].

3 SPO-based opinion mining model

Our algorithm consists of several steps. Firstly, extract all subject-predicate-object triplets on a preprocessed text. Secondly, we use those triplets to build a topic model. Finally, we train the topic model on a corpus of documents and split them into several clusters corresponding opinions. Pic.1 shows a general plan of the algorithm. Next we will be observing every step in detail.



Picture 1: Algorithm scheme

3.1 Extracting SPO triplets

The most basic SPO triplets are noun-verb-noun. Besides explicit triplets there are some implicit examples that do not contain a verb, such as noun-noun triplets. In this work we extract the following types of triplets:

- Explicit triplets: noun-verb-noun.
  Example: *the congress passed a law* $\rightarrow$ *(congress,passed,law)*
- Noun-noun triplets.
  Example: *president Putin* $\rightarrow$ *(Putin,is,president)*
- Adverb triplets.
  Example: *The meeting held by Navalny* $\rightarrow$ *(Navalny, hold, meeting)*
- Adjectives triplets.
  Example: *A new law* $\rightarrow$ *(law,is,new)*

Most works that work with triplets extract them with the Standford NLP Parser, which builds a syntactic tree of a sentence. We too had to build it, but as the NLP Parser does not support Russian we used SyntaxNet, a tensorflow neural network that can build a syntactic tree of a sentence.

3.2 ARTM topic model

In the current section we will describe the basics of additive regularization of topic models. Let D be the corpus of documents and W the set of tokens in them. We will consider every document a bag of words. If each word relates to some topic from T them the corpus is an i.i.d. $(w_i, d_i, t_i)_{i=1}^n$ from a distribution

$p(w,d,t) \in W \times D \times T$. Having presumed that the appearance of a word w in a document depends only on the topic we can draw:

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \varphi_{wt}\theta_{td} = \Phi \cdot \Theta$$

Φ and Θ are stochastic topic-word and document-topic matrices, our goal is to find them. To do that we will be maximizing the likelihood logarithm with probability distribution constraints:

$$\begin{array}{ll} \min\limits_{\Phi,\Theta} & L(\Phi,\Theta) + R(\Phi,\Theta) = \ln \prod\limits_{d \in D} \prod\limits_{w \in d} p(w|d)^{n_{dw}} + R(\Phi,\Theta) = \sum\limits_{d \in D} \sum\limits_{w \in d} n_{dw} \sum\limits_{t \in T} \varphi_{wt}\theta_{td} + R(\Phi,\Theta) \\ \text{s.t.} & \sum_{t \in T} \theta_{td} = 1, \theta_{td} \geq 0 \\ & \sum_{w \in W} \varphi_{wt} = 1, \varphi_{wt} \geq 0 \end{array}$$

Here $R(\Phi,\Theta)$ is the regularizator we add to the likelihood. If $R(\Phi,\Theta) = 0$ it turns into a PLSA model. In this case the problem has an infinite number of solutions, so regularizers provide additional constraints. $R(\Phi,\Theta)$ is usually a sum of regularizers with coefficients: $\sum \tau_i R_i(\Phi,\Theta)$. We will be using the following regularizers in out work:

- Smooth/sparse regularizer:
  We presume that every document and every word relates to a small number of topics, so the distributions $\varphi_t = (\varphi_w t)_{w \in W}, \theta_d = (\theta_{td})_{t \in T}$ should be sparse. At the same time there are some common vocabulary topics that are present in every document, their distributions are smooth. Common vocabulary does not carry much information for out experiments, so we add a smooth regularizer for those words to be gathered in common vocabulary topics. Other topics will then have sparse distributions where only a few words are assigned high probabilities. Those words can be called kernel words.

- Decorrelator regularizer:
  In opinion mining it is necessary that the opinions found in the text corpus are different, otherwise we cannot classify the texts.

3.3 Assigning opinions

Having obtained the matrix Θ with topic distributions for every document we have to assign some opinion to every text, that is to clusterise them. In order to do that we will consider θ_d as a vector of features for d. We can say that feature x_i shows "how much" of topic t is contained in d. To clusterise we used the regular k-means algorithm.

4 Evaluation

To evaluate our algorithm we have used two corpuses of news:

1. 100 news considering enterprise nationalization in LNR and DNR. The texts average at 200 words and they were extracted from multiple news sources: Russian as well as Ukranian. There were three opinions: Moscow's opinion, Kiev's opinion and a neutral opinion. We consider 100 news enough for the evaluation as after that the texts started repeating themselves.

2. 220 news considering Donald Trump's decision of leaving the Paris Agreement. The text's size once again averaged at 200 words. Three were three opinions found: one of Trump's supporters, those who oppose him (such as Elon Musk's) and the neutral opinion.

The corpus of documents has been marked by two independent evaluators who first read them and decided what opinions existed on given topics and then marked every text with an opinion.

4.1 Lexical models

The natural question a reader can stumble upon is "Why use any topic modelling at all? What if opinions have a lexical difference enough to distinguish them?". That is a fair question and two answer it we will be considering two popular algorithms of getting lexical features:

1. TF-IDF:
   Term frequency shows how often a word is used in a document. Inverse document frequency shows how unique the word is to the overall corpus. So if the opinions could be distinguished based on its lexical features alone the TF-IDF of the words in texts expressing different opinions. That is why we choose to use these features.

2. Mean Word2Vec:
   We have trained a Word2Vec model for each corpus and obtained the feature vectors for each word. Then for every text we compute the mean of vectors of it's words. Word2Vec's feature vectors are meant do differ most for distant terms. That makes us believe that if opinions could be identified by lexical features the mean Word2Vec should differ significantly.

Having obtained these features we performed the same k-means algorithm on them.

4.2 Adjusting hyperparameters

Our model has several hyperparameters:

- Number of topics
  That is the overall number of topics including background topics.

- Number of background topics
  Background topics are those that contain common vocabulary. We will not be considering these topics for further clusterisation. They are meant to reduce the noise from common vocabulary words. To clarify, we set our model with a prevailing number of topics and then when clustering do not take into account the background topics.

- Regularization coefficients
  These coefficients determine how much we sparse or smooth the resulting distributions will be.

- Minimal TF
  It is possible to take into account words with term frequency above a threshold.

In order to find the optimal hyperparameters set we simply created a parameter grid and tried every conbination in a selected range for each hyperparameter. This process would have been irrationally long on big datasets, but as we evaluate on 100 texts this is not an issue.

4.3 Quality metrics

Having clusterised our texts we do not know how to map the true clusters to the modelled clusters. That is why we introduce a pairwise metric that does not depend on that. We take all pairs of texts that are in the same cluster originally and compute what part of them remains the one cluster eventually.

$$PW = \sum_{(d_1,d_2)\in D\times D} \frac{[c'(d_1) = c'(d_2)|\ c(d_1) = c(d_2)]}{n}$$

In the formula above c(d) is the initial opinion cluster of a document and the c'(d) is the modelled opinion cluster of a document. The metric ranges in $[0, 1]$.

4.4 Experiments

I would like to remind that the key advance in our model is the use frequencies of subjects and objects to estimate topic distributions. To test wether this approach provides an increase in clusterisation quality we constructed two models: both are PLSA with smooth, sparse and decorrelation regularizers. The difference is that one was trained with word frequencies and the other with subject and object frequencies. Our hypothesis is that using the second approach for approximation of distributions is better. The second hypothesis is that using probabilistic approaches, topic modelling (TM) in particular is better.

	PW
TF-IDF	0.41
Word2Vec mean	0.42
TM on words	0.54
TM on triplets	0.61

Table 1: LNR and DNR enterprises

	PW
TF-IDF	0.42
Word2Vec mean	0.39
TM on words	0.51
TM on triplets	0.58

Table 2: Trump leaving the Paris Agreement

Based on these experiments we can make several conclusions:

1. Texts expressing different opinions have lexical difference, but it is not significant. Algorithms based on TM show better performance on given corpuses.

2. Using triplets instead of words in TM provides an increase in quality.

Of course, these statements can not be fully approved and require extensive research, but based on given data we can make these conclusions.

5 Results and future plans

From this experiment we can draw that using SPO triplets for estimating topic distributions provides an increase in quality. Let us try to explain this result in simple terms. When counting word frequencies in text we can learn what the author mentions. When a person expresses an opinion on some events he usually describes the it focusing on aspects most important for him. Counting how many times a word is used as a subject shows us "how much" the author speaks about it.

Plans for the nearest future include concluding the same sort of experiments on another dataset and trying SPO approach on different opinion and topic mining models.

To construct a completed model for opinion mining it is necessary to use other features, except subject and object frequencies, such as emotive words. After obtaining consistent results with SPO triplets we will move to adding other features.

6 References

[1] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. "Topic sentiment mixture: Modeling facets and opinions in weblogs". In *Proceedings of the World Wide Conference* (2007), pages 171-180.

[2] B. Pang,L. Lee: "Opinion Mining and Sentiment Analysis. Foundations and Trends". In *Information Retrieval* (2008), pages 1-135

[3] M.J. Paul, R. Girju: "Cross-Cultural Analysis of Blogs and Forums with Mixed-Collection Topic Models". In *Proc. of EMNLP 09*(2009), pages 1408-1417

[4] Y. Fang, L. Si, N. Somasundaram, Z. Yu: "Mining Contrastive Opinions on Political Texts using Cross-Perspective Topic Model". In: *Proc. of WSDM 12* (2012), pages 63-72

[5] R. Balasubramanyan, W. W. Cohen, D. Pierce, D. P. Redlawsk "Modeling Polarizing Topics: When Do Different Political Communities Respond Differently to the Same News?" (2012)

[6] M.J. Paul, R. Girju "A Two-Dimensional Topic-Aspect Model for Discovering Multi-Faceted Topics". In *Proc. of AAAI 10* (2010), pages 545-550

[7] X. Zhu, D. Klabjan, P.N. Bless "Unsupervised Terminological Ontology Learning based on Hierarchical Topic Modeling". In *In Proc. of ACL 17* (2017)

[8] E.I.Bolshakova, K.V.Vorontsov and others: "Automatic word processing in natural language and data analysis" (2017), pages 195-228