Hierarchical Viewpoint Discovery from Tweets Using Bayesian Modelling

Lixing Zhu[a], Yulan He[b], Deyu Zhou[a,*]

[a]*School of Computer Science and Engineering, Southeast University, China*
[b]*Department of Computer Science, University of Warwick, UK*

Abstract

When users express their stances towards a topic in social media, they might elaborate their viewpoints or reasoning. Oftentimes, viewpoints expressed by different users exhibit a hierarchical structure. Therefore, detecting this kind of hierarchical viewpoints offers a better insight to understand the public opinion. In this paper, we propose a novel Bayesian model for hierarchical viewpoint discovery from tweets. Driven by the motivation that a viewpoint expressed in a tweet can be regarded as a path from the root to a leaf of a hierarchical viewpoint tree, the assignment of the relevant viewpoint topics is assumed to follow two nested Chinese restaurant processes. Moreover, opinions in text are often expressed in un-semantically decomposable multi-terms or phrases, such as '*economic recession*'. Hence, a hierarchical Pitman-Yor process is employed as a prior for modelling the generation of phrases with arbitrary length. Experimental results on two Twitter corpora demonstrate the effectiveness of the proposed Bayesian model for hierarchical viewpoint discovery.

Keywords: Natural language processing, Opinion mining, Bayesian modelling

1. Introduction

Stance classification aims to predict one's stance in a two-sided debate or a controversial hot topic and has been intensively studied in recent years (Hasan and Ng, 2013; Elfardy et al., 2015). However, apart from detecting one's stance, we are more interested in figuring out the reasons or key viewpoints why the person supports or opposes an issue of interest. Moreover, viewpoints expressed by different users could be related and exhibit a hierarchical structure. Figure 1 illustrates an example hierarchical viewpoint tree, in which both *User A* and *User B* are supporters of Trump. However, the former just expressed his support for Trump without mentioning any reasons behind, while the latter stated the reason that Trump is a charismatic leader. We can also see that both *User C* and *User D* support Trump due to his economic policy, however with different reasons ('*higher employment rate*' vs. '*trade protection*'). Such a hierarchical viewpoint tree enables a better understanding of user opinions and allows a quick glimpse of reasons behind users' stances.

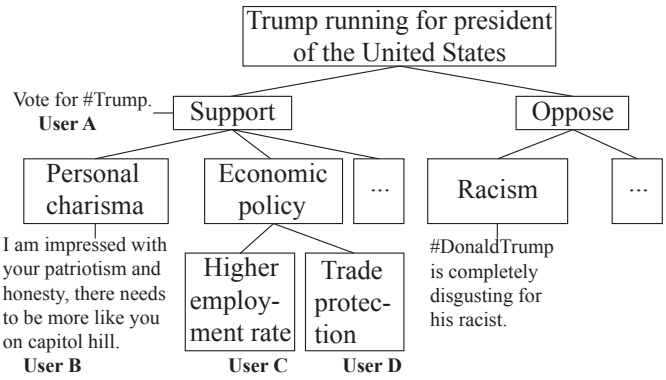


Figure 1: An example of a hierarchical viewpoint tree on the topic "*Trump run for election*" from Twitter.

Mining hierarchical viewpoints from tweets is challenging for the reasons below: (1) In comparison with that the stance is either '*Support*' or '*Oppose*', the hierarchical structure of viewpoints is unknown a priori; (2) People tend to express their opinions in many different ways and even with informal or ungrammatical language; (3) Opinion expressions often contain multi-word phrases, for example, '*economic recession*' and '*economic growth*'. Simply decomposing phrases into unigrams may lose their original semantic meaning. As such, simply applying a bag-of-words topic model might wrongly group them under the same topic due to the shared word '*economic*'.

To tackle these challenges, in this paper, we propose a Bayesian model, called Hierarchical Opinion Phrase (HOP) model, for hierarchical viewpoint discovery from text. In such a model, the root node (level-1) contains the topic of interest (e.g., '*Trump run for president*') and the level-2 topics indicate stance (e.g., either '*Support*' or '*Oppose*'), while topics in the level-3 and below contain viewpoints under different stances. Assuming that viewpoints in each tweet are generated from a path from the root to a leaf of a hierarchical viewpoint tree, the assignment of viewpoint topics can be regarded as following two nested Chinese Restaurant Processes (nCRPs). Furthermore, a hierarchical Pitman-Yor process is employed as a prior to model the generation of phrases with

*Corresponding author. Fax.: 8602552090861.
Email addresses: zhulixing@seu.edu.cn (Lixing Zhu), y.he@cantab.net (Yulan He), d.zhou@seu.edu.cn (Deyu Zhou)

arbitrary length. We have also explored various approaches for incorporating prior information such as sentiment lexicons and hashtags in order to improve the stance classification accuracy. To the best of our knowledge, our work is the first attempt for hierarchical viewpoint detection. The proposed approach has been evaluated on two Twitter corpora. Experimental results demonstrate the effectiveness of our approach in comparison with existing approaches for hierarchical topic detection or viewpoint discovery. Our source code is made available at https://github.com/somethingx86/HOP.

2. Related Work

In this section, we give a brief review of four related lines of research: opinion mining based on topic models, hierarchical topic extraction, topical phrase models and deep learning for sentiment analysis.

2.1. Opinion Mining based on Topic models

Topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) have been proven effective for opinion mining. Lin and He (2009) proposed a Joint Sentiment Topic (JST) model that extends the standard LDA by adding a sentiment layer on top of the topic layer to allow the extraction of positive and negative topics. A variant of JST called reverse-JST was studied in (Lin et al., 2010), where the generation of sentiments and topics is reversed. Kawamae (2012) separated words into aspect words, sentiment words and other words. Aspect words were generated dependent on latent aspects which were in turn sampled from sentiment-associated topics. Kim et al. (2013) modified the aforementioned model by using a recurrent Chinese Restaurant Process (rCRP) prior on the aspect variable. They mined a hierarchy of aspects from product reviews and associated each aspect with a positive or negative sentiment.

In addition to online reviews, the LDA-based models have also been applied on debate forums and Twitter. Lim et al. (2014) proposed a Twitter opinion topic model which makes use of target-opinion pairs. Trabelsi et al. (2014) designed a joint topic viewpoint model by assuming that the distribution over viewpoints is associated with latent topics. Thonet et al. (2016) treated nouns as topic words while adjectives, verbs, adverbs were treated as opinion words for topic-specific opinion discovery. Vilares and He (2017) focused on detecting perspectives from political debates. They modelled topics and their associated perspectives as latent variables and generated words associated with topics or perspectives by following different routes. However, the aforementioned models are not able to generate a hierarchy of opinions.

2.2. Hierarchical Topic Extraction

In general, there are two types of approaches for extracting topical hierarchies. The first one is based on probabilistic graphical models. For example, the Hierarchical LDA (HLDA) model was proposed for discovering the topical hierarchies from abstracts of scientific papers (Blei et al., 2010). In the model, each document is assumed to be attached to a path where each level is a topic. The path will induce a set of topics, and words will be generated in the same way as in LDA. The allocation of paths follows an nCRP prior. Kim et al. (2012) argued that topics should be distributed over the whole nodes of hierarchy. They achieved this by placing an rCRP prior on the tree. Jordan et al. (2015) proposed a nonparametric model called nested hierarchical Dirichlet process to allow shared groups among clusters. Their work extended nCRP by incorporating a hierarchical Dirichlet process.

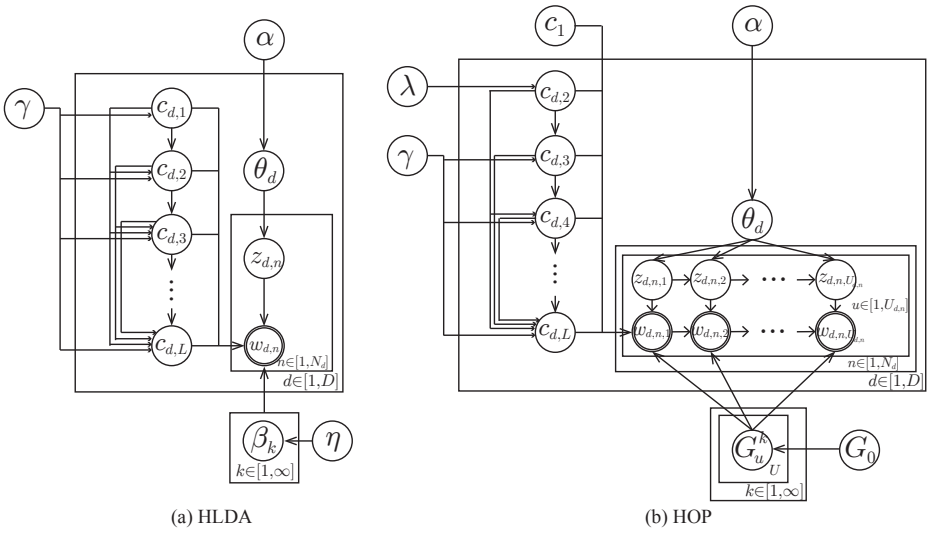
The second type of approaches to hierarchical topic extraction is based on frequent pattern mining. An early work is frequent itemset-based hierarchical clustering model (Fung et al., 2003) which simply clustered documents according to their shared items. Wang et al. (2013) developed a phrase mining framework called CATHY (Constructing A Topical HierarchY). It first builds a term co-occurrence network using a frequent pattern mining method which is commonly used in association rule mining. The initial network corresponds to the root topic. Then the network is clustered into subtopic networks in a probabilistic way by assuming that each term co-occurrence was generated by a topic. The process is repeated until no subtopics can be found.

2.3. Topical Phrase Models

In order to address the wide occurring of un-semantically decomposable phrases, various topical phrase models have been proposed in the literature. Wallach (2006) made the first attempt to extend LDA by incorporating hierarchical Dirichlet language model called Bigram Topic Model (BTM), where each word is generated from a distribution over vocabulary following a two-level hierarchical Dirichlet process. Wang et. al (2007) extended BTM by adding a switch variable at each word position to decide whether to begin a new n-gram or to continue from the previously identified n-gram. El-Kishky et al. (2014) developed a pipeline approach, called TopMine. It first extracts frequent phrases using a frequent pattern mining method. Then an LDA-based model is learned where words in the same phrase are generated from the same topic. He (2016) proposed a topical phrase model which extends TopMine by using the hierarchical Pitman-Yor Process (HPYP) to model the generation of words in a phrase.

2.4. Deep Learning for Sentiment Analysis

Recent years have seen a surge of interests of developing deep learning approaches for sentiment analysis. Many of them have been applied to sentiment classification on product reviews (Chen et al., 2016; Gui et al., 2017), news articles (Lai et al., 2015; Nguyen et al., 2017) as well as tweets (Ghiassi et al., 2013; Dong et al., 2014). Different neural network architectures have been explored including Convolutional Neural Network (CNN) models (dos Santos and Gatti, 2014; Severyn and Moschitti, 2015a,b), Long Short-Term Memory (LSTM) Networks (Tang et al., 2015; Wang et al., 2016) and models with attention mechanisms (Ren et al., 2016; Yang et al., 2016). However, these models rely on annotated datasets where each



(a) HLDA (b) HOP

Figure 2: Graphical models of HLDA and the proposed Hierarchical Opinion Phrase (HOP) model. Boxes are plate notations representing replicates.

Table 1: Notations Used in the Article.

Symbol	*Description*
β_k	Topic k, which is a distribution over the vocabulary
$c_{d,l}$	The lth level, whose value indexes a topic and follows CRP
$\boldsymbol{c}_d$	The path in the tree for tweet d, which follows nCRP
θ_d	Distribution over the levels for tweet d
α	Parameter for the Dirichlet distribution
γ	Concentration parameter for the CRPs
η	Parameter for the symmetric Dirichlet distribution
λ	Parameter for the Bernoulli distribution
$w_{d,n}$	The token in tweet d, position n (HLDA)
$z_{d,n}$	Level allocation for token $w_{d,n}$ (HLDA)
$w_{d,n,u}$	The token in tweet d, phrase n, position u (HOP)
$z_{d,n,u}$	Level allocation for token $w_{d,n,u}$ (HOP)
G_0	Base distribution for the HPYPs
G^k	Topic k, which is an HPYP
G^k_u	The PYP that generates the uth token in phrases of topic k

document is either labeled by the sentiment class or annotated with more fine-grained opinion targets/words for training. As such, they cannot be used for hierarchical opinion discovery in the absence of annotated data.

2.5. Summary

Our model is partly inspired by HLDA. While the root-level topics in HLDA mostly contain background words, our model is able to extract the key topic of interest from tweets at the root-level. Also, the level-2 topics in our model are constrained to be stance-related topics to allow for the incorporation of prior information as will be shown in the experiments section. Furthermore, we incorporate the hierarchical Pitman-Yor Process (HPYP) as the prior to deal with the generation of multi-word phrases and hence the proposed model can generate hierarchical viewpoints with better interpretability.

3. Hierarchical Opinion Phrase (HOP) Model

In this section, we propose the Hierarchical Opinion Phrase (HOP) model to learn a viewpoint hierarchy from text and also model the generation of phrases at the same time. Before presenting the details of HOP, we first describe the HLDA model. The notations used in our model and HLDA are summarized in Table 1.

HLDA (Blei et al., 2010) as illustrated in Figure 2(a) assumes that each topic is tied to a node in a tree and the documents are generated by first selecting a path in the tree, then choosing a topic at each level of the path and finally drawing words from the assigned topics. Notations c in Figure 2(a) are random variables indexing topics β. The per-tweet path $\boldsymbol{c}_d = \{c_{d,1}, c_{d,2}, \ldots, c_{d,L}\}$ follows nCRP, whose valuing behavior functions as the affiliated tweet's seating process, which will be described later in the introduction of nCRP. The arrows linking c indicate the constraint that the path's lower-level node could only take the value of indices of the topics in the restaurant the upper level is pointing to, i.e., a tweet only counts those tweets whose upper level takes the same value in the nCRP of the generative process.

One problem of applying HLDA for viewpoint discovery is its unconstrained number of topics and the mixture of topics under different stances. The problem is aggravated when the corpus is noisy. We modify HLDA by placing a root topic that is shared by all documents to generate common words. Since the number of stances is fixed in our data ("*Support*" or "*Oppose*"), a stance latent variable that has only two possible values is placed at level-2. The stance level will therefore hopefully separate two sets of opposing viewpoints. Since the number of level-2 topics is fixed at 2, it is now possible to incorporate side information into the model such as hashtags indicating the stances, which will be shown in the Experiments section. Another problem with the original HLDA is that it operates on the bag-of-words assumption. As such, the topic results are less interpretable since many phrases are not semantically decomposable. We therefore propose to generate phrases from the modified HLDA model by incorporating the HPYP into the generative process. HPYP has been previously explored for single-level topical phrase extraction in newswire stories and clinical

documents (Lindsey et al., 2012; He, 2016). But it has never been explored for hierarchical topical phrase extraction.

For comparison, the HOP model is illustrated in Figure 2(b) in which the root topic (level-1 topic) contains the topic of interest that is shared across all documents and the level-2 is a stance layer that has only two possible topics, either '*Support*' or '*Oppose*'. Topics in level-3 and below capture viewpoints under different stances.

The proposed approach assumes that phrases have been identified prior to model learning. Phrase extraction can be done by many different approaches. In this paper, we extract phrases from data using an open source toolkit called Gensim.models.phrases[1]. It first discovers candidate phrases based on word collocation patterns, then transforms the phrases into distributed representations, and finally filters out irrelevant phrases (Mikolov et al., 2013). In the following subsections we will discuss the HOP model in more details.

3.1. Generative Process

Suppose there is an L-level hierarchical opinion tree T as shown in Figure 1 where L is fixed, and each tweet contains L latent topics corresponding to a path from the root node to a leaf node. For example, for the tweet "*I am impressed with your patriotism and honesty, there needs to be more like you on capitol hill.*", its hierarchical topics are [TRUMP RUN FOR ELECTION, SUPPORT, PERSONAL CHARISMA]. The root node c_1 of T is shared by all tweets in the collection. The second level of T is limited to have two topics corresponding to two stances, '*Support*' or '*Oppose*', which is assumed to follow a Bernoulli distribution parameterized by λ. The value of lower levels follows an nCRP prior, which can depict the unbound nature of viewpoints. We use $\mathbf{c}_d = \{c_{d,1}, c_{d,2}, \ldots, c_{d,L}\}$ to denote the path assigned to the dth tweet. Therefore, the prior for each level can be expressed as $c_{d,2} \sim \text{Bernoulli}(\lambda)$, $c_{d,l} \sim \text{CRP}(\gamma, c_{d,l-1}, \ldots, c_{d,2})$.

nCRP. We briefly describe the nCRP (Blei et al., 2010) here. The nCRP is employed as the prior for the assignment of topics, which can organize the topics into a tree topology. As illustrated in Figure 3, tweets are analogous to customers and topics are analogous to tables. Assuming that the nth customer enters the root restaurant, he will choose an existing table β_i proportional to the number of customers already sitting there or choose a new table proportional to the concentration parameter γ, that is

$$p(\text{occupied table } i|\text{other customers}) = \frac{n_i}{n-1+\gamma},$$
$$p(\text{new table } i|\text{other customers}) = \frac{\gamma}{n-1+\gamma},$$

where n_i is the number of customers seated at table β_i, n is the total number of customers including the present one, and γ is the concentration parameter normally set to 0.5, which indicates how likely the customer will sit in a new table. After choosing the table β_i, the customer will proceed to choose a table in the lower-level restaurant which is pointed to by Table β_i. This process will continue until the customer reaches depth L. As a result, a path composed of tables (or topics) from L different levels will be induced. This forms L hierarchical topics.

HPYP. We detect phrases based on word collocations so that phrase detection is separated from topic inference. In HOP, phrases will be assigned with the same topic if they are assigned with the same level and their involved tweets happen to share the same table in that level. We use an HPYP to model the generation of phrases assigned with the same topic. HPYP was first proposed in (Teh, 2006), which has been proven to be the best smoothing method for n-gram language models. In HPYP, the distribution of the first word is generated from PYP defined as

$$G_1 \sim \text{PYP}(a_0, b_0, G_0),$$

where G_0 is a uniform distribution over a fixed vocabulary, a_0 is the discount parameter indicating the degenerating rate of to-generate power law distribution and b_0 is the concentration parameter controlling the amount of variability of G_1 around G_0. The distribution of the second word is generated using G_1 as the base distribution, $G_2 \sim \text{PYP}(a_1, b_1, G_1)$. The process continues until the end of the phrase. The generative process of drawing words from the prior can be simulated by the generalised CRP (Pitman et al., 2002). A restaurant corresponds to each G_u, where each table is served a dish whose value is chosen from the base distribution G_{u-1}. The first customer sits at the first table; the $n+1$th customer chooses an occupied table in proportion to the number of customers already sitting there and takes the value of dish on the table, or chooses a new table proportional to a constant parameter and orders a dish from the base distribution. This process continues until the proxy customer sits in an existing table or there is no parent restaurant. As such, the probability of the uth word given the seating arrangement is

$$p(w|\Lambda_u) = \frac{C_{uw} - a_{u-1}T_{uw}}{C_{u\cdot} + b_{u-1}} + \frac{a_{u-1}T_{u\cdot} + b_{u-1}}{C_{u\cdot} + b_{u-1}} \times p(w|\Lambda_{u-1}),$$

where C_{uw} is the number of customers having dish w in the restaurant u, T_{uw} denotes the number of tables serving dish w in restaurant u, $C_{u\cdot} = \sum_w C_{uw}$ and $T_{u\cdot} = \sum_w T_{uw}$.

It is worth noting that the restaurant setup here is different from that in nCRP. In nCRP each tweet is a customer and each topic is a table. A tweet is assigned with L topics (or tables) from the root to the leaf. In HPYP for phrase generation, each word w is a customer and a restaurant u is the context of the word. For example, for an n-gram phrase, the context of the nth word is its preceding $n-1$ words.

Based on the above description, the generative process of HOP is given below.

- For each topic $k \in \{1, 2, 3, \ldots, \infty\}$:
  - $G_1^k \sim \text{PYP}(a_0, b_0, G_0)$
  - $G_2^k \sim \text{PYP}(a_1, b_1, G_1^k)$
  - $\cdots$
  - $G_U^k \sim \text{PYP}(a_{U-1}, b_{U-1}, G_{U-1}^k)$
- Set c_1 to be the root restaurant
- For each tweet $d \in \{1, 2, 3, \ldots, D\}$:

[1] http://radimrehurek.com/gensim/models/phrases.html#id1

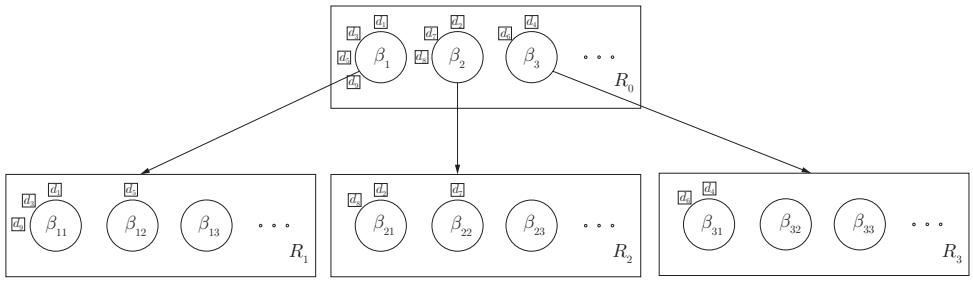


Figure 3: Illustration of the nested Chinese Restaurant Process (nCRP). Each circle represents a table which points to a unique restaurant denoted as a rectangle. Each customer (or tweet) will first choose a table in the upper level then follow the link pointed to by the table to reach a lower-level restaurant and choose another table there.

- Select the Level-2 topic $c_{d,2} \sim \text{Bernoulli}(\lambda)$
- For level $l \in \{3, \ldots, L\}$, select its corresponding topic $c_{d,l} \sim \text{CRP}(\gamma, c_{d,l-1}, \ldots, c_{d,2})$
- Draw a distribution over levels $\theta_d \sim \text{Dirichlet}(\boldsymbol{\alpha})$
- For each phrase $n \in \{1, 2, \ldots, N_d\}$:
  - For each word $u \in \{1, 2, \ldots, U_{d,n}\}$:
    If it is the first word in a phrase ($u = 1$):
    - Assign a level $z_{d,n,u}|\theta_d \sim \text{Discrete}(\theta_d)$
    - Draw a word $w_{d,n,u}|\{z_{d,n,u}, \mathbf{c}_d, \mathbf{G}\} \sim \text{Discrete}(G_1^{c_d[z_{d,n,u}]})$

    Else:
    - Set $z_{d,n,u} = z_{d,n,u-1}$
    - Draw a word $w_{d,n,u}|\{z_{d,n,u}, \mathbf{c}_d, \mathbf{G}\} \sim \text{Discrete}(G_u^{c_d[z_{d,n,u}]})$.

Here, $c_d[z_{d,n,u}]$ denotes the $z_{d,n,u}$th component of vector $\mathbf{c}_d$, G_0 is a uniform distribution over a fixed vocabulary $\mathcal{W}$ of V words, for $\forall w \in \mathcal{W}$, $G_0(w) = \frac{1}{V}$.[2] Since phrases have already been identified prior to hierarchical opinion extraction, the boundaries of phrases are observed and need not be sampled from data.

3.2. Inference and Parameter Estimation

Given the observed variable $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D\}$, our goal is to infer the hidden variables $\mathbf{c}$ and $\mathbf{z}$ using the posterior distribution $p(\mathbf{c}, \mathbf{z}|\mathbf{w}, \lambda, \gamma, \boldsymbol{\alpha}, G_0, \Omega)$ where $\Omega = \{a_0, b_0, \ldots, a_U, b_U\}$ denotes the hyper-parameters of HPYPs. Since exact inference of the posterior distribution is intractable, Gibbs sampling (Griffiths and Steyvers, 2004) is employed to approximate the hidden variables, which sequentially samples each variable of interest using the conditional probability of that variable given the current values of all other variables and the data. With sufficient iterations, the sampling process will finally reach a status when all the samples can be seen as generated by a stationary distribution.

The variables used in the sampling process are: (1) $z_{d,n,u}$, the level allocation of the uth word in the nth phrase of the dth tweet; (2) $\mathbf{c}_d$, the path of the dth tweet. The objective of the Gibbs sampler is to approximate $p(\mathbf{c}, \mathbf{z}|\mathbf{w}, \lambda, \gamma, \boldsymbol{\alpha}, G_0, \Omega)$. We omit Ω for clarity. In Gibbs sampling, we are focused on the per-document posterior $p(\mathbf{c}_d, \mathbf{z}_d|\mathbf{c}_{-d}, \mathbf{z}_{-d}, \mathbf{w}, \lambda, \gamma, \boldsymbol{\alpha}, G_0)$. For $d \in \{1, 2, \ldots, D\}$, we first sample word-wise level allocation $p(z_{d,n,u}|\mathbf{c}, \mathbf{z}_{-d,-n,-u}, \mathbf{w}, \boldsymbol{\alpha}, G_0)$, where $\mathbf{z}_{-d,-n,-u}$ is the vector of level allocations leaving out $z_{d,n,u}$. Then we sample path $p(\mathbf{c}_d|\mathbf{c}_{-d}, \mathbf{z}, \mathbf{w}, \lambda, \gamma, G_0)$.

3.2.1. Level Allocation Sampling

Given a path, we need to sample level allocations for the tweet d. According to Bayes' rule and conditional independence, we obtain

$$\begin{aligned} p(z_{d,n,u} = l|\mathbf{c}, \mathbf{z}_{-d,-n,-u}, \mathbf{w}, \boldsymbol{\alpha}, G_0) \propto \\ p(z_{d,n,u} = l|\mathbf{z}_{d,-n,-u}, \boldsymbol{\alpha}) \times \\ p(\mathbf{w}_{d,n}|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-d,-n}, G_0). \end{aligned} \tag{1}$$

The first term in Eq. 1 is a conditional probability marginalizing out θ_d. Since the Dirichlet distribution $p(\theta_d|\boldsymbol{\alpha})$ and the discrete distribution $p(z_{d,n,u}|\theta_d)$ form a Dirichlet-Multinomial conjugate, we have

$$p(z_{d,n,u} = l|\mathbf{z}_{d,-n,-u}, \boldsymbol{\alpha}) = \frac{C^l_{d,-n,-u} + \alpha[l]}{C^{\cdot}_{d,-n,-u} + \sum_{l=1}^{L} \alpha[l]}.$$

Here, $C^l_{d,-n,-u}$ is the number of times level label l being assigned to some word tokens in tweet d leaving out $z_{d,n,u}$, $C^{\cdot}_{d,-n,-u} = \sum_{l=1}^{L} C^l_{d,-n,-u}$, L is the total depth of the hierarchy.

The second term in Eq. 1 follows a $z_{d,n,u}$-specified HPYP with $\mathbf{w}_{d,n}$ as a new random variable given the particular status of that HPYP. We use the generalised CRP to perform sampling. Let $\Lambda^{c_d[l]}$ denote the current seating arrangement of topic $c_d[l]$, the second term is rewritten as $p(\mathbf{w}_{d,n}|\Lambda^{c_d[l]})$. Thereafter, the random process can be simulated by the situation that the first word $w_{d,n,1}$ enters the restaurant $\Lambda_1^{c_d[l]}$ as a customer. The probability of the next word w from $G_u^{c_d[l]}$ can be calculated recursively as

$$p(w|\Lambda_u^{c_d[l]}) = \frac{C_{uw}^{c_d[l]} - a_{u-1}T_{uw}^{c_d[l]}}{C_{u\cdot}^{c_d[l]} + b_{u-1}} + \frac{a_{u-1}T_{u\cdot}^{c_d[l]} + b_{u-1}}{C_{u\cdot}^{c_d[l]} + b_{u-1}} \times p(w|\Lambda_{u-1}^{c_d[l]}).$$

Here, $C_{uw}^{c_d[l]}$ denotes the number of customers eating dish w in the restaurant u owned by topic $c_d[l]$, $T_{uw}^{c_d[l]}$ denotes the number of tables serving dish w in restaurant u owned by topic $c_d[l]$,

[2] Note that we abuse the notation here that we use n to denote the nth phrase.

$C_{u\cdot}^{c_d[l]} = \sum_w C_{uw}^{c_d[l]}$ and $T_{u\cdot}^{c_d[l]} = \sum_w T_{uw}^{c_d[l]}$. The recursion ends when $u = 1$, that is

$$p(w|\Lambda_1^{c_d[l]}) = \frac{C_{1w}^{c_d[l]} - a_0 T_{1w}^{c_d[l]}}{C_{1\cdot}^{c_d[l]} + b_0} + \frac{a_0 T_{1\cdot}^{c_d[l]} + b_0}{C_{1\cdot}^{c_d[l]} + b_0} \frac{1}{V}.$$

If $w_{d,n,u}$ is not the first word of a multi-term phrase, we just take $z_{d,n,u} = z_{d,n,u-1}$ and do not sample a new topic.

3.2.2. Path Sampling

Given paths of other tweets and level allocations, we have to sample the path for tweet d. By applying the Bayes' rule, we have

$$p(\mathbf{c}_d|\mathbf{c}_{-d}, \mathbf{z}, \mathbf{w}, \lambda, \gamma, G_0) \propto p(\mathbf{c}_d|\mathbf{c}_{-d}, \lambda, \gamma) \times p(\mathbf{w}_d|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-d}, G_0). \tag{2}$$

The first term in Eq. 2 is a prior over paths. It can be computed by first calculating a Bernoulli distribution then calculating each level's seating distribution in the corresponding restaurant.

The second term in Eq. 2 is the probability of a given tweet under a possible seating arrangement in the nCRP tree. It can be decomposed into probabilities of phrases/words occurred in the tweet,

$$\begin{aligned} p(\mathbf{w}_d|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-d}, G_0) = p(\mathbf{w}_{d,1}|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-d}, G_0)\times \\ p(\mathbf{w}_{d,2}|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-d}, \mathbf{w}_{d,1}, G_0)\times \\ \dots \\ p(\mathbf{w}_{d,N_d}|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-d}, \mathbf{w}_{d,1}, \dots, \mathbf{w}_{d,N_d-1}, G_0). \end{aligned} \tag{3}$$

Each term in Eq. 3 is a posterior distribution of phrase $\mathbf{w}_{d,n}$ conditioned on other phrases allocated with the same topic. The distribution follows an HPYP, and thus can be sampled in the same way as described for the second term in Eq. 1.

3.2.3. Complete Sampling Procedure

Algorithm 1 Gibbs sampling for HOP

1. Initialize the model by arbitrarily assigning a path to each tweet. Randomly assign a level number of the path to each word/phrase in the tweet. Initialize the HPYP configuration within each topic for each associated word/phrase.
2. For each tweet $d \in \{1, 2, \dots, D\}$:
   (a) Sample $\mathbf{c}_d^{(t+1)}$ using Eq. 2.
   (b) For each phrase n in tweet d, $n \in \{1, 2, \dots, N_d\}$:
      i. For each word u in phrase n, $u \in \{1, 2, \dots, U_{d,n}\}$:
         A. Sample $z_{d,n,u}^{(t+1)}$ using Eq. 1.
3. Repeat step 2 until the global log-likelihood converges or a fixed number of iterations is reached.
4. Output the final sample $\{\mathbf{c}, \mathbf{z}\}$.

Given the conditional distributions defined above, we are able to perform the full Gibbs sampling. Let $\{\mathbf{c}^{(t)}, \mathbf{z}^{(t)}\}$ denote the current state, the sampling process is described in Algorithm 1.

4. Experiments

In this section, we first describe the datasets and baselines used in our experiments. We then evaluate HOP against the baselines quantitatively and qualitatively.

4.1. Experimental Setup

Algorithm 2 Opinion tweets retrieval.

1. Define the seed patterns "*[brexit|leaving the EU|staying in the EU]+ [would|can|might|won't|can't]*" for Dataset I (Brexit), and "*[Trump |Hillary|Donald Trump|Hillary Clinton]+[would|can|might|won't|can't]*" for Dataset II (US General Election). These seed patterns are used for retrieving seed tweets such as "*Top economic think tanks agree that brexit would harm economy. Vote Remain in the EU*" or "*Trump would start WW3! There is no one he hasn't offended*". We denote the set of the seed patterns as $\mathcal{P}$ and the set of opinion tweets as $\mathcal{T}$.
2. (a) Perform Part-of-Speech (POS) tagging on the seed tweets $\mathcal{T}$.
   (b) Extract keywords which are tagged as the latter noun in the POS tag pattern "*NN+MD+VB+NN*".
   (c) Enlarge the seed pattern set $\mathcal{P}$ by adding new rules based on the newly extracted keywords such as "*keyword+[would|can|might|won't|can't]*".
   (d) Retrieve tweets based on the enlarged seed pattern set. Add the retrieved tweets to $\mathcal{T}$.
3. Repeat step 2 until no more tweets are found.

To evaluate HOP, two datasets are constructed. We developed a crawler[3] using the Twitter4j toolkit and Twitter Streaming API[4]. The crawler listens to the global timeline and filters tweets with specific keywords. Dataset I contains 515, 113 tweets crawled using hashtags such as *#EURef*, *#EU referendum* and *#Brexit*, dated 14th-24th June 2016. Dataset II contains 1, 691, 294 tweets with hashtags *#PresidentElection*, *#Election2016*, *#Hillary*, *#Trump*, dated 1st-8th November 2016. To ensure tweets with opinions are kept, a rule-based method inspired by (Handler et al., 2016) is used and is described in Algorithm 2. The process runs iteratively until no more tweets are found. The statistics of the final datasets are shown in Table 2. It can be noticed that only 13.33% of tweets are kept in Dataset I and for Dataset II the proportion is 2.13%.

Each tweet is pre-processed by removing common stop words. No stemming is used. Phrases are identified based on collocation patterns from data using Gensim[5]. We perform Gibbs sampling for a maximum of 10,000 iterations and output the intermediate results every 1,000 iterations. It usually takes

[3] https://github.com/somethingx86/EclipseTwitterStreamer
[4] https://developer.twitter.com/en/docs
[5] http://radimrehurek.com/gensim/models/phrases.html

Table 2: Statistics of the two datasets.

Dataset	Property	Value
I	#tweets	68,672
	#unigram tokens	1,219,484
	vocabulary size	31,149
II	#tweets	36,013
	#unigram tokens	591,620
	vocabulary size	21,595

around 1,000 iterations to reach a stationary status when the topic topology and tweet-topic associations no longer change, which can be visualized using a tree structure.

4.2. Methods for Comparison

We compare our model with the following approaches which can generate topic hierarchies:

CATHY (Constructing A Topical HierarchY) (Wang et al., 2013) builds a topical hierarchy where each topic is represented as a set of phrases. A term co-occurrence network is first constructed using the Frequent Pattern (FP)-growth algorithm. The network edges are clustered according to their associated topics to obtain a hierarchy of topics. In our experiments, the hierarchy depth is set to 3, the cluster number is set to 2 for level 2 and 4 for level 3. Other parameter settings follow (Wang et al., 2013).

HLDA (Hierarchical LDA) (Blei et al., 2010) assumes that each document is generated by drawing an infinite L-level path according to the nCRP prior, and drawing a topic distribution over levels in the path according to a stick-breaking process. Words are drawn from the L topics which are associated with the restaurants along that path.

HASM (Hierarchical Aspect-Sentiment Model) (Kim et al., 2013) produces hierarchical aspects in which each aspect is associated with positive, negative or neutral polarities. The aspect hidden variable follows an rCRP. We use the default parameter settings in our experiments. Note that the prior sentiment knowledge from some common sentiment seed words is used to set the asymmetric Dirichlet prior for aspect-sentiment-word distributions in HASM. In our experiments, its default sentiment lexicon is replaced by SENTIMENT140LEX[6] (Mohammad et al., 2013) which was specifically constructed for Twitter sentiment analysis.

For HOP, the parameter settings are: $\lambda = 0.5$, $\gamma = 0.5$, $\boldsymbol{\alpha} = [0.75, 0.15, 0.075, 0.025]$, $a_u = 0.8$, $b_u = 1$ for HPYP, the maximum phrase length $U = 3$. We only keep topics which are associated with at least 1,000 tweets.

4.3. Topic Coherence

Various measures have been proposed to evaluate the quality of the topics discovered. Newman et al. (2010) found that the pointwise mutual information (PMI) of all word pairs in a topic's top ten words coincides well with human judgements. PMI is defined as follows:

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, \quad (4)$$

where $p(w_i, w_j)$ is the co-occurrence likelihood of two words. It can be estimated by counting the co-occurrence of the word pair in sliding windows in an external large meta-document. Röder et. al (2015) studied the known coherence measures and proposed a new measure which was a combination of some existing ones. Particularly, this metric first retrieves co-occurrence counts for the given words using a sliding window of size 110 in Wikipedia. For each top word a vector is built whose components are the normalized Point-wise Mutual Information (PMI) between the word and every other top words. The arithmetic mean of all vector pairs' cosine similarity is treated as the coherence measure of a given topic.

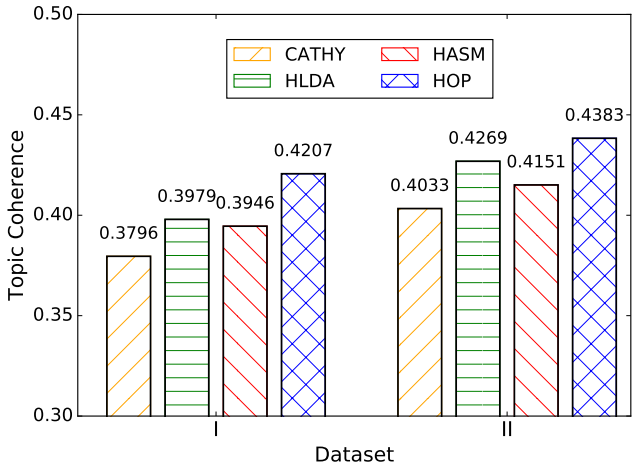


Figure 4: Topic coherence on two datasets.

Following the measure proposed in (Röder et al., 2015), we report the average topic coherence scores computed on the top 10 words/phrases of each topic which is shown in Figure 4. It can be observed that CATHY scores the lowest compared to the other three methods on both datasets. HLDA gives better results compared to HASM. HOP outperforms all the other methods and the improvement over the second best performing model, HLDA, is more prominent on Dataset I.

4.4. Stance Classification

Some previous studies used topic models to perform sentiment/stance classification and achieved comparable results (Trabelsi and Zaıane, 2014; Thonet et al., 2016). Apart from HLDA and HASM, we select two more baselines which can output document-level stance labels:

VODUM (Viewpoint and Opinion Discovery Unification Model) (Thonet et al., 2016) assumes nouns are topical words and adjectives, verbs and adverbs are opinion words. It uses different generative routes to generate topical words and opinion words. Their model associates a viewpoint label (equivalent to a stance label here) with each document.

[6] http://www.saifmohammad.com/Lexicons/Sentiment140-Lexicon-v0.1.zip

sLDA (Supervised LDA) (Mcauliffe and Blei, 2008) modified LDA by adding an observed response variable to each document, which follows a Gaussian distribution whose mean is the averaged weighed sum of topic assignments of all the document's tokens.

To obtain the ground-truth for the evaluation of stance classification, we hired three senior undergraduate students, who worked on NLP-related final-year projects, to manually annotate some randomly selected tweets from Dataset II. Tweets were discarded if there were disagreement among the annotators. In total, 80 tweets were discarded. In the end, we kept a total of 1,000 tweets which consists of 748 positive and 252 negative tweets. We compare HOP with baselines on this dataset for stance classification. For HOP, we use the level-2 topics for stance classification. For HLDA, there is no restriction on its level-2 topic number and we manually go through all the topics in level-2 to identify the likely '*Support*' and '*Oppose*' stances. Afterwards, each document's stance can be identified accordingly. For VODUM, the number of viewpoints (stances) is set to 2 and all the hyperparameters take the default setting. We run each approach for 20 times and average the classification results over 20 such runs.

HASM used the prior sentiment knowledge to set the asymmetric Dirichlet prior for aspect-sentiment-word distributions. In more details, it placed higher probabilities on some polarity seed words based on a given sentiment lexicon. The incorporation of this kind of supervised information gave them an edge over the unsupervised alternatives. In order to compare our model with HASM fairly, we also experimented with a variant of HOP (call 'pHOP') by incorporating the prior information of stance and sentiment into the model. The prior information can be derived from either hashtags or existing sentiment lexicons. For hashtags, we use '*#VoteTrump*' and '*#NeverHillary*' as a proxy label for the '*Support*' stance, and '*#VoteHillary*' and '*#NeverTrump*' as a proxy label for the '*Oppose*' stance. In the experiments, the prior information was only utilized during model initialization, where the '*Support*' and '*Oppose*' tweets were assigned to the left and right path in Level-2, respectively. Furthermore, we also utilize the sentiment prior information of words which is obtained from SENTIMENT140LEX. We consider those with sentiment intensity larger than 1.0 as strong sentiment words. We restrict those words with strong sentiment from SENTIMENT140LEX to be only generated from the Level-2 topics. During model initialization, if a tweet has a stance label or a word token can be found in the sentiment lexicon, then the path or level allocation will be kept. Otherwise, a path or level allocation will be randomly initialized. There are about 10% tweets carrying proxy labels, and 1.15% words found in the sentiment lexicon in our data. For sLDA, proxy stance labels are treated as the observed response variables to train the model.

Stance classification results are presented in Figure 5. HLDA, VODUM and HOP are all unsupervised approaches without the prior information. It can be observed that HLDA gives better results compared to VODUM but VODUM tends to be more stable. HOP outperforms both HLDA and VODUM. Although the best performance achieved is only about 58%, this is still noticeable given that HOP is totally unsupervised.

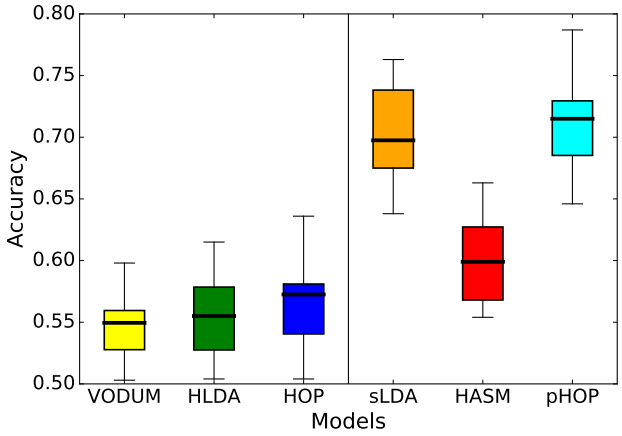


Figure 5: Stance classification accuracy.

By incorporating the prior information into the model, we notice a significant boost in stance classification accuracy where pHOP achieves an accuracy of 71% and it outperforms sLDA by 2%. HASM only achieves an accuracy of 60% since it only utilizes the sentiment lexicon and cannot benefit from the proxy stance labels due to its unbounded aspect node arrangement for each sentence, which has to follow an rCRP.

4.5. Qualitative Results

To evaluate the quality of opinion hierarchies discovered, we compare HOP with CATHY and HLDA. Figure 6 illustrates the opinion hierarchies generated by these three models respectively on Dataset I. It can be observed that CATHY can distinguish two opposing stances at level-2, but it generated less coherent topics at level-3. Oftentimes, CATHY tends to group phrases sharing common constituent words into the same topic, e.g., Topic 2 and 4 under the right branch of the level-2 topic. HLDA generates more sensible topics at level-3. However, it is not able to distinguish two stances at level-2 since the number of topics is unconstrained. Also, top topic words listed at level-3 for HLDA are dominated by unigrams and are less interpretable. This is not surprising since HLDA operates under the bag-of-phrases assumption. Phrases such as '*immigration policy*' and '*immigration control*' are treated as two distinct tokens although they share the same constituent word '*immigration*'. Hence, running HLDA on data with phrases identified suffers from more severe data sparsity as the vocabulary size is significantly enlarged.

On the contrary, we can observe that HOP generates more distinct opinions[7], especially at level-2. It can also be observed that HOP has more phrases appeared in its level-3 topics compared to HLDA. Overall, HOP offers better interpretability of the topics discovered. Moreover, when examining the hierarchical topic assignment to each individual tweet, we notice that

[7]We manually add the topic labels in bold face for easy understanding of the results.

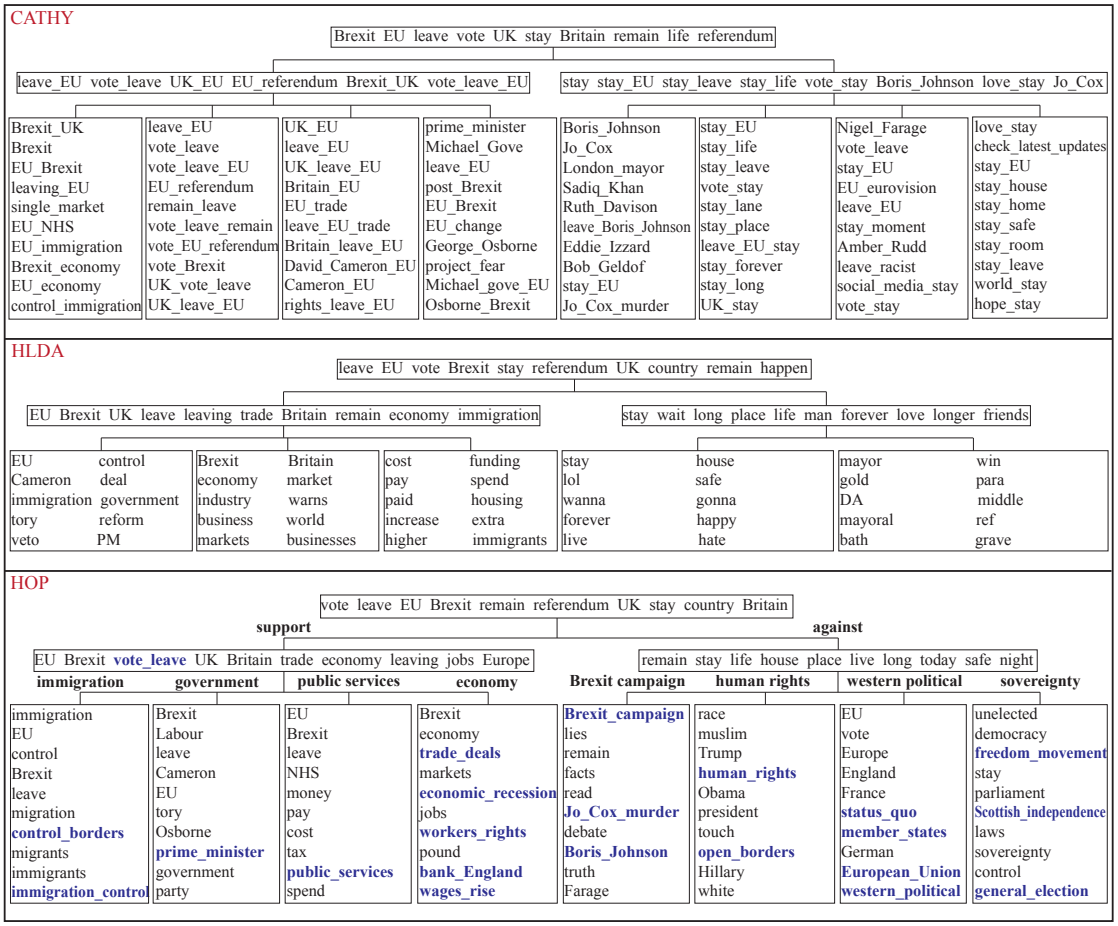


Figure 6: Opinion hierarchies discovered by CATHY, HLDA and HOP on Dataset I.

in many cases, tweets are assigned with sensible hierarchical opinions. For example, "*Briefing: Trade, investment and jobs will benefit if we Vote Leave*" is assigned with [SUPPORT, ECONOMY]. It can be easily inferred from the topic result that the tweet is about the support of Brexit because of an economy-related reason.

On Dataset II only the hierarchy generated by HOP is presented in Figure 7 due to the space limit. Although the level-2 topics do not clearly represent two opposing stances since the HOP model is totally unsupervised, it can be inferred from the level-3 topics ('*Email scandal of Hillary*' and '*Trump vowed to drain the swamp*[8]') that the left level-2 topic is about '*Supporting Trump*'. And similarly we can infer from the level-3 topics that the right level-2 topic is about '*Supporting Hillary*'. Also, under the '*Drain the swamp*' topic, the 4th level gives more fine-grained topics on '*Controlling illegal immigrants*' and '*Trump will lead a unified Republican government*'.

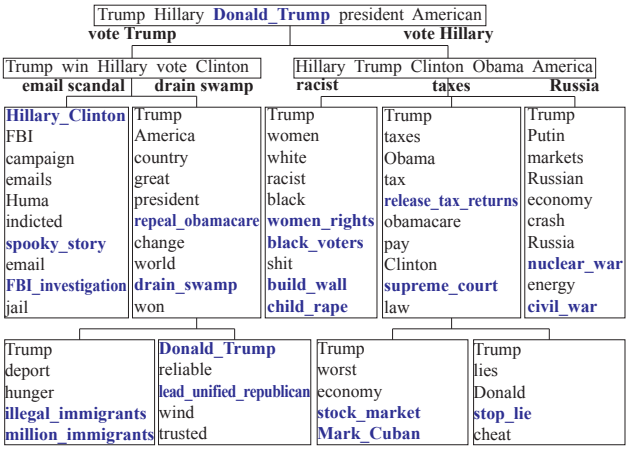


Figure 7: Opinion hierarchy discovered by HOP on Dataset II.

5. Conclusion

In this paper, we have proposed an unsupervised Hierarchical Opinion Phrase (HOP) model in which each document is associated with a path in a topic tree and with a document-specific distribution over different levels in the tree. Phrases are drawn from their associated topic-specific HPYP. Experimental results on two Twitter datasets show that our proposed HOP model is able to reveal hierarchical opinions from social media. It also shows that HOP significantly outperforms existing approaches to hierarchical topic extraction in both topic

[8]Trump used this metaphor to describe his plan to fix problems in the federal government.

coherence and stance classification. In our current work, all the paths in the generated hierarchical opinion tree have the same depth. In future work, we will explore modelling hierarchical opinion trees with varying depths of path.

Acknowledgements

We would like to thank the reviewers for their valuable comments and helpful suggestions. This work was funded by the National Natural Science Foundation of China (61528302, 61772132), the Natural Science Foundation of Jiangsu Province of China (BK20161430) and Innovate UK (grant no. 103652).

References

Blei, D. M., Griffiths, T. L., and Jordan, M. I. (2010). The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016). Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., and Xu, K. (2014). Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 49–54.

dos Santos, C. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

El-Kishky, A., Song, Y., Wang, C., Voss, C. R., and Han, J. (2014). Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment*, 8(3):305–316.

Elfardy, H., Diab, M., and Callison-Burch, C. (2015). Ideological perspective detection using semantic features. *Lexical and Computational Semantics (* SEM 2015)*, page 137.

Fung, B. C., Wang, K., and Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 59–70. SIAM.

Ghiassi, M., Skinner, J., and Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, 40(16):6266–6282.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235.

Gui, L., Zhou, Y., Xu, R., He, Y., and Lu, Q. (2017). Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowledge-Based Systems*, 124:34–45.

Handler, A., Denny, M. J., Wallach, H., and OâĂŹConnor, B. (2016). Bag of what? simple noun phrase extraction for text analysis. *NLP+ CSS 2016*, 114.

Hasan, K. S. and Ng, V. (2013). Frame semantics for stance classification. In *CoNLL*, pages 124–132.

He, Y. (2016). Extracting topical phrases from clinical documents. In *AAAI*, pages 2957–2963.

Kawamae, N. (2012). Hierarchical approach to sentiment analysis. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 138–145. IEEE.

Kim, J. H., Kim, D., Kim, S., and Oh, A. (2012). Modeling topic hierarchies with the recursive chinese restaurant process. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 783–792. ACM.

Kim, S., Zhang, J., Chen, Z., Oh, A. H., and Liu, S. (2013). A hierarchical aspect-sentiment model for online reviews. In *AAAI*.

Lai, S., Xu, L., Liu, K., and Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Lim, K. W. and Buntine, W. (2014). Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1319–1328. ACM.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM.

Lin, C., He, Y., and Everson, R. (2010). A comparative study of bayesian models for unsupervised sentiment detection. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 144–152. Association for Computational Linguistics.

Lindsey, R. V., Headden III, W. P., and Stipicevic, M. J. (2012). A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 214–222.

Mcauliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In *Advances in neural information processing systems*, pages 121–128.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Mohammad, S., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics.

Nguyen, D., Vo, K., Pham, D., Nguyen, M., and Quan, T. (2017). A deep architecture for sentiment analysis of news articles. In *International Conference on Computer Science, Applied Mathematics and Applications*, pages 129–140. Springer.

Paisley, J., Wang, C., Blei, D. M., and Jordan, M. I. (2015). Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270.

Pitman, J. et al. (2002). Combinatorial stochastic processes.

Ren, Y., Zhang, Y., Zhang, M., and Ji, D. (2016). Context-sensitive twitter sentiment classification using neural network. In *AAAI*, pages 215–221.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM.

Severyn, A. and Moschitti, A. (2015a). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM.

Severyn, A. and Moschitti, A. (2015b). Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 464–469.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Teh, Y. W. (2006). A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics.

Thonet, T., Cabanac, G., Boughanem, M., and Pinel-Sauvagnat, K. (2016). Vodum: a topic model unifying viewpoint, topic and opinion discovery. In *European Conference on Information Retrieval*, pages 533–545. Springer.

Trabelsi, A. and Zaıane, O. R. (2014). Finding arguing expressions of divergent viewpoints in online debates. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@ EACL*, pages 35–43.

Vilares, D. and He, Y. (2017). Detecting perspectives in political debates. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1573–1582.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984.

ACM.

Wang, C., Danilevsky, M., Desai, N., Zhang, Y., Nguyen, P., Taula, T., and Han, J. (2013). A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–445. ACM.

Wang, X., McCallum, A., and Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 697–702. IEEE.

Wang, Y., Huang, M., Zhao, L., et al. (2016). Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.