RESEARCH ARTICLE

Extractive and Abstractive Sentence Labelling of Sentiment-bearing Topics

Mohamad Hardyman BARAWI[1], Chenghua LIN[2], Advaith SIDDHARTHAN[3], Yinbin LIU[4]

1 Computing Science, University of Aberdeen, UK
2 Department of Computer Science, University of Sheffield, UK
3 Knowledge Media Institute, The Open University, UK
4 School of Management, Shanghai University, China

RESEARCH ARTICLE

Extractive and Abstractive Sentence Labelling of Sentiment-bearing Topics

Mohamad Hardyman BARAWI[1], Chenghua LIN [2], Advaith SIDDHARTHAN[3], Yinbin LIU(✉)[4]

1   Computing Science, University of Aberdeen, UK
2   Department of Computer Science, University of Sheffield, UK
3   Knowledge Media Institute, The Open University, UK
4   School of Management, Shanghai University, China

Abstract

This paper tackles the problem of automatically labelling sentiment-bearing topics with descriptive sentence labels. We propose two approaches to the problem, one extractive and the other abstractive. Both approaches rely on a novel mechanism to automatically learn the relevance of each sentence in a corpus to sentiment-bearing topics extracted from that corpus. The extractive approach uses a sentence ranking algorithm for label selection which for the first time jointly optimises topic–sentence relevance as well as aspect–sentiment co-coverage. The abstractive approach instead addresses aspect–sentiment co-coverage by using sentence fusion to generate a sentential label that includes relevant content from multiple sentences. To our knowledge, we are the first to study the problem of labelling sentiment-bearing topics. Our experimental results on three real-world datasets show that both the extractive and abstractive approaches outperform four strong baselines in terms of facilitating topic understanding and interpretation. In addition, when comparing extractive and abstractive labels, our evaluation shows that our best performing abstractive method is able to provide more topic information coverage in fewer words, at the cost of generating less grammatical lables than the extractive method. We conclude that abstractive methods can effectively synthesise the rich information contained in sentiment-bearing topics.

1   Introduction

Probabilistic topic models such as latent Dirichlet allocation (LDA) [1] capture the thematic properties of documents by modelling texts as a mixture of distributions over words, known as topics. The words under each topic tend to co-occur together and consequently are thematically related to one another. These topics can therefore be used as a lens for exploring and understanding large archives of unstructured text. Since the introduction of LDA, many extensions have been proposed, an important one being the Joint Sentiment-Topic (JST) model that aims to mine and uncover rich opinion structures from opinionated documents [2]. This work has spurred subsequent research in developing variants of sentiment topic models for a range of opinion mining tasks such as aspect-based sentiment analysis [3], contrastive opinion mining [4], and the analysis of sentiment and topic dynamics [5].

JST is a hierarchical topic model which can detect sentiment and topic simultaneously from opinionated documents. The hidden topics discovered, therefore, are essentially sentiment-bearing topics resembling opinions. This is a key difference compared to the standard topics extracted by LDA which only express thematic information. We exemplify this difference using two topic examples shown in

Table 1: Examples of LDA and JST topics with nouns (unformatted), adjectives(italics) verbs (bold).

[LDA]: computer models information data system network model *parallel* methods software
[JST]: amazon order **return ship receive** refund *damaged disappointed* policy *unhappy*

Table 1, where each topic is summarised using the ten topic words with the highest marginal probability. By examining the topic words, we see that the terms of the LDA topic are recognisably about the theme *"computer technology"*, whereas the terms of the JST topic capture opinions relating to *"unsatisfactory online shopping experience"*.

Although sentiment topic models have become an increasingly popular tool for exploring and understanding opinions from text, existing models all share some noticeable drawbacks which can significantly limit their usefulness and effectiveness. First, applying sentiment topic models for exploratory purposes requires unfolding the meaning of the topics discovered, which, so far, relies entirely on manual interpretation. However, it is generally difficult for a non-expert user to understand a topic based only on its multinomial distribution, especially when the user is not familiar with the source collection [6]. The second limitation lies in these models' inability to facilitate accurate opinion and sentiment understanding, which is crucial for many applications. The issue stems from the fact that, although the topic words of sentiment-bearing topics collectively express opinions, only a limited understanding of these opinions is possible by examining a list of topic words. Using the JST topic in Table 1 as an example, through manual examination one can interpret that this topic expresses opinions relating to an *"unsatisfactory online shopping experience"*, but it is impossible to gain any deeper insight, e.g. whether the sentiment *unhappy* is being expressed about the product being ordered or about Amazon's policies.

Remarkably, there is no existing work that has studied the problem of automatic labelling of sentiment-bearing topics. Substantial work on automatic topic labelling exists, but has focused on labelling standard topics discovered by LDA to facilitate the interpretation of topics discovered. These works note that standard topics mainly express thematic information [7] and that most of the high ranking topic terms are nouns, and therefore aim to extract *phrasal* labels that capture this thematic association to the topic terms. In contrast, the top terms of sentiment-bearing topics consist of a good mixture of nouns, verbs, and adjectives resembling opinions rather than themes [2, 8–10]. Existing topic labelling approaches engineered for standard LDA topic are intrinsically unsuited to labelling sentiment-bearing topics as they provide

no mechanism for modelling opinionated content.

In this paper, we formally study the problem of automatically labelling sentiment-bearing topics extracted by the JST model [2]. In contrast to existing approaches which generate topic labels in the form of either a single term [11], a small set of terms (e.g. the top-*n* topic words) [12], or phrases [7], we choose sentence as the label modality to facilitate interpretation of the opinions encoded in sentiment-bearing topics. Specifically, we propose two novel approaches for automatically generating sentence labels that can facilitate understanding and interpretation of multinomial sentiment-bearing topics extracted by JST, one extractive and the other abstractive. Both approaches rely on a novel mechanism to automatically learn the relevance of each sentence in a corpus to sentiment-bearing topics extracted from that corpus. The extractive approach uses a sentence ranking algorithm for label selection which for the first time jointly optimises this sentence relevance and aspect–sentiment co-coverage. The abstractive approach provides an alternate solution to aspect–sentiment co-coverage by using sentence fusion to generate a sentential label that includes relevant content from multiple sentences.

We compare the effectiveness of our extractive topic labelling approaches against four strong extractive baselines, including two sentence label baselines and two topic labelling systems which have been widely bench-marked in the literature [6, 7]. Experimental results on three real-world datasets show that the labels generated by our approaches are more effective than all the baselines in terms of facilitating sentiment-bearing topic understanding and interpretation. We then compare our extractive approach to two abstractive approaches and show that abstractive approaches can generate shorter labels with better coverage of the topic, but with some loss of grammaticality. To summarise, our contributions in this paper are three-fold: (i) we introduce a novel mechanism which can automatically learn the relevance to sentiment-bearing topics of the underlying sentences in a corpus; (ii) we design a sentence label selection criteria which jointly considers relevance to the topic as well as aspect and sentiment co-coverage and show that it beats several strong baselines; (iii) we demonstrate the effectiveness of sentence fusion methods for generating abstractive labels which enhance topic coverage by including information from multiple

sentences.

To our knowledge, we are the first to study the problem of labelling sentiment-bearing topics. We describe related work in Section 2, followed by our labelling method in Section 3, experimental design in Section 4, results in Section 5 and conclusions in Section 6.

2 Related Work

Since its introduction, latent Dirichlet allocation (LDA) has become a popular tool for unsupervised analysis of text, providing both a predictive model of future text and a latent topic representation of the corpus. The latter property of topic models has enabled exploration and digestion of large text corpora through the extracted thematic structure represented as a multinomial distribution over words. Still, a major obstacle to applying LDA is the need to manually interpret the topics, which is generally difficult, especially for non-expert users. This has in turn motivated a number of approaches to automatically generating meaningful labels that facilitate topic understanding.

2.1 Automatic Topic Labelling

Automatically learning topic labels from data. In early work, Mei et al. [6] proposed generating topic labels using either bigrams or noun phrases extracted from a corpus and ranked the relevance of candidate labels to a given topic using KL divergence. Candidate labels were scored using relevance functions which minimised the similarity distance between the candidate labels and the topic words. The top-ranked candidate label was then chosen as the topic's label. Mao et al. [13] labelled hierarchical topics by investigating the sibling and parent-child relations among the topics. Their approach followed a similar paradigm to [6], which ranked candidate labels by measuring the relevance of the labels to a reference corpus using Jensen-Shannon divergence. More recently, Cano et al. [12] proposed an approach to labeling topics based on multi-document summarisation. They measured the term relevance of documents to topics and generated topic label candidates based on the summarisation of a topic's relevant documents.
Automatic topic labelling leveraging external sources. Another type of approach in automatic topic labelling leverages external sources, e.g., Wikipedia or DBpedia [11, 14, 15]. Lau et al. [11] proposed selecting the most representative term from a topic as its label by computing the similarity between each word and all others in the topic. Several sources

of information were used to identify the best label, including pointwise mutual information scores, WordNet hypernymy relations, and distributional similarity. These features were combined in a re-ranking model. In a follow-up work, Lau et al. [14] generated label candidates for a topic based on top-ranking topic terms and titles of Wikipedia articles. They then built a Support Vector Regression (SVR) model for ranking the label candidates. Other researchers [15] proposed using graph centrality measures to identify DBpedia concepts that are related to the topic words as their topic labels. More recently, Aletras et al. [7] proposed generating topic labels of multiple modalities, i.e., both textual and images labels. To generate textual label candidates, they used the top-7 topic terms to search Wikipedia and Google to collect article titles which were subsequently chunk parsed. The image labels were generated by first querying the top-n topic terms to the Bing search engine. The top-n candidate images retrieved were then ranked with PageRank and the image with the highest PageRank score was selected as the topic label. More recently, [16] proposed to label topics using a pre-computed dependency-based word embeddings [17], whereas [18] generated topic labels by utilising personalised domain ontologies.

To summarise, the topic labelling approaches in the above-mentioned works were engineered for labelling standard LDA topics rather than sentiment-bearing topics, and mostly fall into one of the following linguistic modalities: a single term [11], a small set of terms (e.g., the top-n topic words) [11], or phrases [7, 14]. These linguistic modalities, while sufficient for expressing facet subjects of standard topics, are inadequate to express complete thoughts or opinions due to their linguistic constraints, i.e. lacking a subject, a predicate or both. Our goal is to address this gap and study the problem of automatically labelling sentiment-bearing topics with more appropriate and descriptive sentence labels.

2.2 Sentence Fusion

Sentence fusion is a text-to-text generation technique that transforms overlapping information from a cluster of similar sentences into a single sentence. One of the most popular strategies for sentence fusion relies on merging the dependency trees of input sentences to produce a graph representation which will be linearised in a separate stage [19–22].

Barzilay and McKeown [19] introduced a sentence fusion framework for multi-document summarisation. Their approach provides a means to capture the main information in a cluster of related sentences in comparison to the com-

mon method which just selects the sentence closest to the centroid. Approaches to sentence fusion typically use dependency graphs to produce an intermediary syntactic representation of the information in a cluster of sentences [19–21]. The graph is then linearised in different ways to generate fused sentences, which are then ranked using a language model or language-specific heuristics to filter out ill-formed sentences [23]. This type of approaches of fusing sentences mimic the strategies used by humans as reported in the analysis of human-generated summaries in [24]. The sentence fusion task has since been widened to encompass other variants such as combining two sentences to produce a single sentence that either conveys the common information shared by the two sentences, or all information of sentences but without redundancy [21, 22, 25–27]. More recent works [28–30] on sentence fusion consider not only pairs of sentences but also larger sentence clusters using *multi-sentence compression*, where the compression is performed by selecting the high-scoring paths in a weighted bigram graph [28–30].

3   Methodology

The goal of our work is to develop automatic approaches for labelling multinomial sentiment-bearing topics with descriptive sentence labels. We first briefly introduce the JST model used for extracting sentiment-bearing topics, and then proceed to describe the proposed approaches for sentence label generation. It should be noted that our approaches do not have any specific dependencies on the JST model, and thus they are general enough to be directly applied to any other sentiment topic model variants that generate multinomial topics as output.

3.1   Preliminaries of the JST Model

The graphical model of JST is shown in Figure 1, in which D denotes a collection of documents, N_d a sequence of words in document d, S the number of sentiment labels, T the total number of topics, and $\{\alpha, \beta, \gamma\}$ the Dirichlet hyperparameters (cf. [2] for details). The formal definition of the generative process in JST corresponding to the graphical model shown in Figure 1 is as follows. First, one draws a sentiment label l from the per-document sentiment proportion π_d. Following that, one draws a topic label z from the per-document topic proportion $\theta_{d,l}$ conditioned on sentiment label l. Finally, one draws a word from the per-corpus word distribution $\varphi_{l,z}$ conditioned on both the sentiment label l and topic label z.
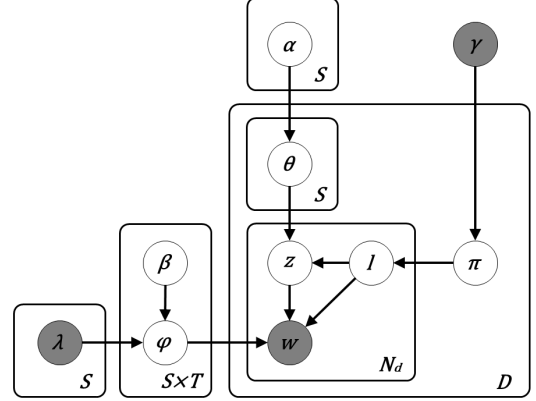

Fig. 1: JST graphical model.

- For each sentiment label $l \in \{1, ..., S\}$
  - For each topic $j \in \{1, ..., T\}$
    * draw $\varphi_{lj} \sim \text{Dir}(\lambda_l \cdot \beta_{lj}^T)$
- For each document $d \in \{1, ..., D\}$,
  - Choose a distribution $\pi_d \sim \text{Dir}(\gamma)$
  - For each sentiment label l under document d
    * Choose a distribution $\theta_{d,l} \sim \text{Dir}(\alpha)$
  - For each word w_i in document d
    * Choose a sentiment label $l_j \sim \text{Mult}(\pi_d)$
    * Choose a topic $z_i \sim \text{Mult}(\theta_{d,l_i})$
    * Choose a word $w_i \sim \text{Mult}(\varphi_{l_i z_i})$

The problem to be solved by the JST model is the posterior inference of the variables, which determine the hidden sentiment-bearing topic structures that can best explain the observed set of documents. Formally, a sentiment-bearing topic is represented by a multinomial distribution over words denoted as $p(w|l, z) \equiv \varphi_{w,l,z}$, where w, l, and z represent the word, sentiment label and topic label indices, respectively. In particular, the approximated per-corpus sentiment-topic word distribution is

$$\varphi_{l,z,w} = \frac{N_{l,z,w} + \beta}{N_{l,z} + V\beta}, \tag{1}$$

where V is the corpus vocabulary size, $N_{l,z,w}$ is the number of times word w appeared in topic z with sentiment label l, $N_{l,z}$ is the number of times words are assigned to z and l. Recalling that $\varphi_{w,l,z}$ is a multinomial distribution, we have $\sum_{w \in V} \varphi_{w,l,z} = 1$.

3.2   Modelling the Relevance between Sentiment-bearing Topics and Sentences

The original JST model can only learn topic–word and topic–document associations as it operates on bag-of-words features at the document-level, with the corpus sentence struc-

ture being ignored. Therefore, we propose a new computational mechanism that can uncover the relevance to a sentiment-bearing topic of the underlying sentences in the corpus. To achieve this, we first preserve the sentence structure information for each document during the corpus pre-processing step (see Section 4 for more details). Second, modelling topic-sentence relevance is essentially equivalent to calculating the probability of a sentence given a sentiment-bearing topic $p(\text{sent}|l, z)$, i.e., the likelihood of a sentence (from the corpus) associating with a given sentiment-bearing topic. The posterior inference of JST, based on Gibbs sampling [2, 31], can recover the hidden sentiment label and topic label assignments for each word in the corpus. Such label-word assignment information provides a means to re-assembling the relevance between a word and a sentiment-bearing topic. By leveraging the sentence structure information and gathering the label assignment statistics for each word of a sentence, we can derive the probability of a sentence given a sentiment-bearing topic as

$$p(\text{sent}|l, z) = \frac{p(l, z|\text{sent}) \cdot p(\text{sent})}{p(l, z)}$$
$$\propto p(l, z|\text{sent}) \cdot p(\text{sent}), \qquad (2)$$

where

$$p(l, z|\text{sent}) = \frac{\sum_{\substack{w \in \text{sent} \vee \\ l'_w = l, z'_w = z}} \varphi_{l', z', w}}{\sum_{w \in \text{sent}} \varphi_{l', z', w}} \qquad (3)$$

$$p(\text{sent}) = \sum_l \sum_z \prod_{w \in \text{sent}} \varphi_{l, z, w}. \qquad (4)$$

Note that the per-corpus sentiment-topic word distribution $\varphi_{w,l,z}$, defined in Eq. 1, is obtained via the posterior inference using Gibbs sampling. Also $p(l, z)$ is discounted as it is a constant when comparing sentencial labels for the same sentiment-bearing topic. A summary of our mechanism of calculating $p(\text{sent}|l, z)$ is given in Algorithm 1.

3.3 Extractive labelling of sentiment-bearing topics

Given a sentiment-bearing topic, one intuitive approach for selecting the most representative sentence label is to rank the sentences in the corpus according to the topic-sentence relevance probability $p(\text{sent}|l, z)$ derived in the previous section. One can then select the sentence with the highest probability as a label for the topic. We found that a high-degree of relevance, however, cannot be taken as the only criterion for label selection as it might result in the selection of short sentences that do not provide enough information about the topic or that cover either thematic or sentiment information alone. To address these issues and to select the most representative

---

Algorithm 1 Procedures of calculating $p(\text{sent}|l, z)$.

Input: A corpus C
Output: The probability of observing a sentence given a sentiment-bearing topic $p(\text{sent}|l, z)$.

1: Perform automatic sentence segmentation on corpus C to preserve the sentence structure information.
2: Train a JST model based on C following the standard procedures and settings described in [2].
3: **for** each sentiment label $l \in \{1, ..., S\}$ **do**
4:    **for** each topic label $z \in \{1, ..., K\}$ **do**
5:       **for** each sent $\in C$ **do**
6:          Calculate $p(\text{sent}|l, z)$ based on Eq. 2
7:       **end for**
8:       Normalise the probability of $p(\text{sent}|l, z)$
9:       Sort sentences based on the probability of $p(\text{sent}|l, z)$ in a descending order.
10:    **end for**
11: **end for**

---

sentence labels covering sentiment-coupled aspects which reveal opinions, it is important to consider the occurrences of both aspect and sentiment, and the balance between them. To this end, we designed a sentence label selection criterion which jointly considers relevance as well as aspect and sentiment co-coverage. This provides the basis for building an algorithm for ranking sentence labels in terms of how well they can describe the opinion encoded in a sentiment-bearing topic.

Given a sentiment-bearing topic, we define the sentence scoring function below:

$$L(s|t_{l,z}) = \alpha \cdot Rel(s|t_{l,z}) + (1 - \alpha) \cdot Cov(s|t_{l,z}). \qquad (5)$$

Here, $Rel(s|t_{l,z})$ is the relevance score between a sentence s and a given sentiment-bearing topic $t_{l,z}$ as described in Section 3.2. $Cov(s|t_{l,z})$, the aspect and sentiment co-coverage score, encodes two heuristics: (i) sentence labels covering either sentiment or aspects information alone will be significantly down-weighted; and (ii) labels which cover salient aspects of the topic coupled with sentiment will be given high weightage. Parameter α sets the relative contributions of relevance and the co-coverage of aspect and sentiment, and was empirically set to 0.4. Let i be a word from sentence s, a_i a binary variable that indicates whether word i is an aspect word, o_i a binary variable that indicates whether word i is a sentiment word, $w_{l,z,i}$ the importance weight of word i given topic $t_{l,z}$, and $t_{l,z,n}^i$ a binary variable which indicates the presence of word i in the top-n words of sentiment-bearing topic $t_{l,z}$. Formally, the aspect and sentiment co-coverage score is formulated as

$$Cov(s|t_{l,z}) = \frac{2 \cdot A(s|t_{l,z}) \cdot S(s|t_{l,z})}{A(s|t_{l,z}) + S(s|t_{l,z})} \qquad (6)$$

$$A(s|t_{l,z}) = \sum_{i \in s} w_{l,z,i} \cdot a_{s,i} \cdot t^i_{l,z,n} \qquad (7)$$

$$S(s|t_{l,z}) = \sum_{i \in s} w_{l,z,i} \cdot o_{s,i} \cdot t^i_{l,z,n}, \qquad (8)$$

where the weight $w_{l,z,i}$ essentially equals to $\varphi_{l,z,i}$, i.e., the marginal probability of word i given a sentiment-bearing topic $t_{l,z}$. We identify whether word i is an aspect word or a sentiment word as follows. In the preprocessing, we perform parts of speech (POS) tagging on the experimental datasets (detailed in Section 4). Words tagged as nouns are regarded as aspects words whereas sentiment words are the ones that have appeared in the MPQA sentiment lexicon[1]. Recalling that a multinomial topic is represented by its top-n topic words with the highest marginal probability, we further constrain that word i must also appear in the top-n topic words of sentiment-bearing topic $t_{l,z}$. This ensures that a good sentence label should cover as many important sentiment coupled aspects of a topic as possible. In the case where a sentence contains either aspect or sentiment information alone (i.e., $A(s|t_{l,z}) = 0$ or $S(s|t_{l,z}) = 0$), the resulting aspect and sentiment co-coverage score would be zero, thus down-weighting the corresponding sentence. Finally, a sentence label $\hat{s}$ for a sentiment-bearing topic $t_{l,z}$ can be obtained by:

$$\hat{s} = \underset{s}{\operatorname{argmax}}\, L(s|t_{l,z}). \qquad (9)$$

3.4 Abstractive topic labelling with multi-sentence compression

Our second type of approach to topic labelling employs sentence fusion techniques for generating abstractive labels. We hypothesise that abstractive sentence labels would be more succinct by removing redundant information of input sentences, while maintaining the essence of sentiment-bearing topics as much as possible. Specifically, we explore two well established sentence fusion algorithms [28, 29] for the fusion task, both of which search for an optimal weighting function in noisy graphs to identify readable and informative compressions.

3.4.1 Word graph-based multi-sentence compression

Our first sentence fusion approach (named **PathGraph**) extends the algorithm of Filippova [28] to produce well punctuated and informative compression. Taking the sentences extracted in Section 3.2 as input, we build a word graph from a set of related sentences $S = \{s_1, s_2, ..., s_n\}$, by iteratively adding sentences to it. Figure 2 is an illustration of the word

---

graph constructed from the four sentences below. For clarity, edge weights are omitted, and italicised fragments from the sentences are replaced with dots.

1. It was easy to set up and use.
2. Easy manual and set up.
3. It was *a super* easy setup.
4. *The D-Link DP-300U* was very easy to set up.

The word graph is constructed as follows. First, the first sentence is added to the graph represented by a string of word nodes plus the start and the end symbols, i.e., the `start` and `end` nodes in Figure 2. A word from the remaining sentences is mapped onto an existing node in the graph if they have the same lowercased word form and part of speech (POS), in addition to that there is no word from the sentence that is already mapped onto the node. A new node is created if there is no suitable candidate node in the word graph. Words are added to the graph in the three steps: (i) non-stopwords for which no candidate exists in the graph or for which a non-ambiguous mapping is possible; (ii) non-stopwords for which there are either several possible candidates in the graph or which appear more than once in the sentence; and finally (iii) stopwords.

In the original algorithm of Filippova [28], punctuation marks are excluded [28]. We add an additional step to generate well-punctuated compressions by adding punctuation marks in the graph following [29], and select the candidate that has the same immediate context if the mapping is ambiguous. Once all the words from a sentence are added to the graph, words adjacent in the sentence are connected with directed edges, where the edge weights are calculated using the weighting function defined in Eq. 10 and Eq. 11.

$$w(i, j) = \frac{\text{coherence}(i, j)}{\text{freq}(i) \times \text{freq}(j)} \qquad (10)$$

$$\text{coherence}(i, j) = \frac{\text{freq}(i) + \text{freq}(j)}{\sum_{s \in S} \text{dist}(s, i, j)^{-1}} \qquad (11)$$

Here $\text{freq}(j)$ is the number of words mapped to the node j. The function $\text{dist}(s, i, j)$ refers to the distance between the offset positions of words i and j in sentence s. The purpose of this function is two-fold: (i) to generate a grammatical compression, it favours links between words which often occur in this order (see Eq. 11); (ii) to generate an informative compression, it promotes paths passing via salient nodes.

To produce more informative sentences which maximise the range of topics they cover, we further make use of the top-15 topic words extracted by JST. The rationale behind is that topic words can capture the gist of documents, and thus
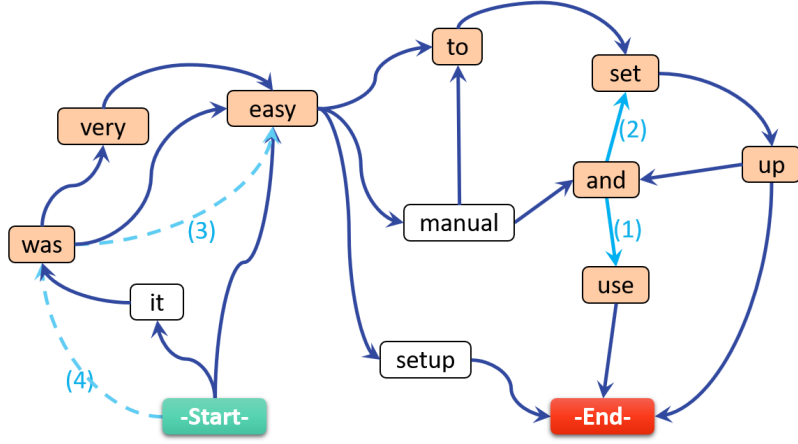
Fig. 2: Word graph constructed from four related sentences where a possible compression path is also given.

can be used to better generate sentences that convey the gist of the set of related sentences. To do so, we added syntactic constraints in the path selection process to filter all the paths which do not contain any topic words (among the top-15 topic words) for the topic being labelled, which do not pass a verb node and which are shorter than eight words. Finally, the remaining paths are re-ranked by normalising the total path weight over its length.

3.4.2 Word graph compression by keyphrases relevance

The second sentence fusion algorithm (named **Keyphrase**) extends the work of [29], which is similar to GraphPath but differs in that it re-ranks the generated compressions according to the number relevant keyphrases a sentence compression contains. Specifically, keyphrases are extracted from the cluster of related sentences in two steps. First, a weighted graph is constructed from the set of related sentences, where nodes represent tuples consisting of lowercased words as well as their corresponding parts of speech. Two nodes are connected if their corresponding lexical units co-occur within a sentence. TextRank [32] is then applied to compute a salience score for each node. The second step generates and scores keyphrase candidates. A candidate keyphrase k is scored by summing the salience scores of the words it contains and normalising by its length (i.e., Eq. 12).

$$\text{score}(k) = \frac{\sum_{w \in k} \text{TextRank}(w)}{\text{length}(k) + 1} \quad (12)$$

During the path selection process, we apply the same syntactic constraints as used for GraphPath, i.e., a candidate path must contain as least one top-15 topic words, one verb node,

and satisfy the path length requirement. Finally, all candidate paths are re-ranked by normalising the total path weight over its compression length multiplied by the sum of the scores of the keyphrases it contains. The score of a sentence compression c is finally given by:

$$\text{score}(c) = \frac{\sum_{i,j \in \text{path}(c)} w_{(i,j)}}{\text{length}(c) \times \sum_{k \in c} \text{score}(k)} \quad (13)$$

where $w_{(i,j)}$ is the edge weight defined in Eq. 10.

4 Experimental Setup

4.1 Dataset

We evaluated the effectiveness of the sentence label selected by our approach on three real-world datasets. Two datasets are publicly available, i.e., Amazon reviews for kitchen and electronic products, where each dataset contains 1000 positive and 1000 negative reviews[2] [33]. We have also collected a movie review dataset for *Internal Affairs* and *The Departed* from IMDb[3]. These two movies virtually share the same storyline but with different casts and productions: *Internal Affairs* is a 2002 Hong Kong movie, while *The Departed* is a 2006 Hollywood movie.

4.2 Implementation Details

In the JST model implementation, we set the symmetric prior β=0.01 [31], the symmetric prior $\lambda = (0.05 \times L)/S$, where L

---

[2] https://www.cs.jhu.edu/ mdredze/datasets/sentiment/
[3] http://www.imdb.com/

Table 2: Dataset statistics.

Dataset	total # of doc.	avg. # of words	avg. # of sentences	Vocab. size
Kitchen	2,000	25	4	2,450
Electronics	2,000	25	4	2,317
IMDb	1,383	118	13	14,337

is the average of the document length, S the is total number of sentiment labels, and the value of 0.05 on average allocates 5 percent of probability mass for mixing. The asymmetric prior α is learned directly from data using maximum-likelihood estimation [34] and updated every 50 iterations during the Gibbs sampling procedure. We empirically set the model with 2 sentiment labels and 10 topics for each sentiment label, resulting 20 sentiment-bearing topics in total for each of the three evaluation datasets. In the preprocessing, we first performed automatic sentence segmentation[4)] on the experimental datasets in order to preserve the sentence structure information of each document, followed by parts of speech tagging on the datasets using the Stanford POS tagger [35]. Punctuation, numbers, and non-alphabet characters were then removed and all words were lowercased. Finally, we trained JST models with the preprocessed corpus following the standard procedures described in [2], with the MPQA lexicon [36] being incorporated as prior information for model learning.

4.3 Baselines

Top probability The first baseline (**Top-prob**) is to select the sentence with highest topic relevance probability $p(\text{sent}|l, z)$ according to the JST model, as described in Section 2.

Centroid. The second baseline is a **centroid** based sentence label. For each sentiment-bearing topic, we first construct a sentence cluster consists of the top 150 most relevant sentences for the given topic, ranked based on the topic relevance probability $p(\text{sent}|l, z)$. Next, for each of the sentence in the cluster, we compute the cosine similarity between the sentence and the remaining sentences in the cluster. Finally, the sentence with the highest cosine similarity score is picked as the topic label. This is a stronger baseline as it selects a sentence representative of multiple high probability sentences.

Phrase/bigram baselines. In addition to the two sentential baselines, we also compare our approach to two systems which have been widely bench-marked in the topic labelling task, namely, **Mei07** [6] and **Aletras14** [37]. Note that **Mei07** is our re-implementation which generates bigram labels, while **Aletras14** is the original implementation kindly

---

Table 3: Human evaluation: Top-1 average quality ratings.

Domain	Mei07	Aletras14	Top-prob	Centroid	Sent-label
Kitchen	0.31	1.29	0.64	1.79	1.81
Electronics	0.52	2.13	0.83	1.33	2.48
IMDb	0.36	1.73	0.69	1.67	2.58
Average	0.39	1.72	0.72	1.59	2.29

Table 4: Average label length in words across three datasets.

Mei07	Aletras14	Top-prob	Centroid	Sent-label
6	6.80	6.83	23.80	12.70

provided by the authors which generates phrase labels. For both, we selected the top three bigrams/phrases for inclusion in the label.

---

5 Experimental Results

We evaluated our extractive topic labelling approach through an experiment where participants rated labels generated by our approaches as well as two baselines and two competitor systems in Section 5.1. For the evaluation of our abstractive labelling approach, we added an additional Grammaticality measure as one of the evaluations for the label generated using the abstractive approach. The description of the evaluation is detailed in Section 5.2.

5.1 Evaluation of extractive sentence labelling

Method. For each sentiment-bearing topic, annotators were provided with the top 10 topic words with the highest marginal probability and the labels generated by each system. They were then asked to judge the quality of each label on a 4-point Likert Scale:

- 3 Very good label, a perfect description of the topic;
- 2 Reasonable label, but does not completely capture the topic;
- 1 Label is semantically related to the topic, but would not make a good topic label; and
- 0 Label is completely inappropriate, and unrelated to the topic.

The most relevant review of the sentiment-bearing topic (calculated using $p(d|l, z)$, the probability of a document given a sentiment-bearing topic) was also provided to annotators so that they can have a better context for topic under-

---

[4)] http://www.nltk.org/

standing. To ensure there was no bias introduced in this step, we checked whether the most relevant review contains our sentence label. This only occurred in 7.5% of cases. For the remaining 92.5%, sentence labels appeared in only the second to the fifth most relevant reviews, not shown to the user. This suggest that our experimental setup is not biased towards our system as the most relevant review rarely contains the most representative sentence label.

We recruited 30 Computing Science postgraduate students to participate in the evaluation task. Recalling that 20 sentiment-bearing topics were extracted for each dataset and that there are 5 systems to compare (i.e., our proposed approach and the four baselines) and 3 datasets, the total number of labels to rate is 300. To avoid the evaluation task being overly long, we broke the evaluation task into 3 sessions carried out on different days, with each session evaluating labels from one dataset only. Completing one session took around 18 minutes for each participant on average.

Quantitative Results. We report results using the Top-1 average rating [14], which is the average human rating (based on the Likert Scale) assigned to the best candidate labels selected by each of the approaches, and the higher rating will indicate a better overall quality of a label. Table 3 shows the Top-1 average ratings for each approach (with our approach denoted as **Sent-label**). The sentence labels generated by our approach outperforms all baselines significantly (two-tailed paired t-test; $p < 0.001$ in all cases). Furthermore, as shown in Table 4, the label selected by our approach also keeps a good balance of information-richness and brevity, in contrast to the other two sentential baselines that favour either very short or very long sentences. To measure inter-annotator agreement (IAA), we first calculated Spearman's ρ between the ratings given by an annotator and the average ratings from all other annotators for the labels corresponding to the same topic. We then averaged the ρ across annotators, topics, and datasets, resulting in an average $\rho = 0.73$, a good IAA.

We want to stress that these baselines are fair. To our knowledge, we are the first to study the problem of labelling sentiment-bearing topics, so there is no directly comparable system available which can generate sentence labelling for sentiment-bearing topics. In the past, top-*n* topic terms have been commonly used for manually interpreting sentiment-bearing topics for a variety of sentiment-topic models (i.e., it's not just exclusively used for standard topic models), and that is why we propose an automatic labelling approach here. For the phrase label baselines, they can capture a fair amount of sentiment information from sentiment-bearing topics based on the adjective/adverb phrases extracted. The

sentence label baselines are also strong baselines as they both make use of the probability distributions from the JST model, with the centroid baseline further addressing the diversity issue.

Qualitative analysis. Table 5 shows six sentiment-bearing topics for the Kitchen, Electronics and IMDb dataset extracted by JST. For each topic, we also show the top-3 labels generated by Mei07 and Aletras14, two baselines (i.e., Centroid, Top-prob), and the sentence label generated by our approach (Sent-label). It can be seen from the table that the sentence labels generated by our approach generally capture the opinions encoded in the sentiment-bearing topics quite well; whereas in quite a few cases the labels generated by Mei07 and Aletras14 either only capture the thematic information of the topics (e.g., *Noise-cancelling headphones*) or merely sentiment information (e.g., *very good*).

Our labelling approach also shows better performance than the sentential baselines. For instance, both sentential baselines were unable to adequately interpret the negative sentiment-bearing topics in the Kitchen dataset. For example, the Centroid baseline "*I ordered ... received only the bottom piece.*" captures the thematic information (i.e., *product*) but fails to capture the sentiment information (e.g., *worth*). The Top-prob baseline captures negative sentiment information (i.e., *disappointed*) but does not include any thematic information (e.g., *money, product*). A user who read the Top-prob baseline (i.e., "*Heres why I am sorely disappointed*") is likely to have some confusions in understanding what causes the disappointment in the label, e.g., whether it is due to *money* or *product*. Our approach (i.e., Sent-label) also performs better in the Electronics dataset by giving more descriptive information about products. Take the positive sentiment-bearing topics for example, the labels generated by Mei07 and Aletras14 capture two different thematic aspects i.e., *sound* and *headphones*, respectively. For the Top-prob label, it captures positive sentiment about *sound quality*, whereas the Centroid baseline label seems less relevant to the topic. In contrast, our proposed approach shows that the topic conveys the opinion of iPod sound quality being amazing.

To summarise, our experimental results show that our extractive sentence labelling approach outperforms four strong baselines and demonstrates the effectiveness of our sentence labels in facilitating topic understanding and interpretation.

5.2 Evaluation of abstractive sentence labelling

Method. To evaluate the effectiveness of the abstractive labels, we compared the compressions generated by GraphPath

Table 5: Labelling examples for sentiment-bearing topics.

		Positive Topics	Negative Topics
Kitchen	**Topic words**	*very good well really nice price quality set little happy*	*money dont product disappointed back waste without very worth make*
	Mei07	are very; a very; very good	the money; money on; you will
	Aletras14	prices and quality; nice guy; coin grading	food waste; zero waste; waste your money
	Top-prob	Well worth every penny.	Heres why I am sorely disappointed:
	Centroid	I got this for the dogs and they seem to like it.	I ordered the product and when I opened the box I found that I received only the bottom piece.
	Sent-label	Nice little toaster with good price.	Totally dissatisfy with this product and it's damn waste of money.
Electronics	**Topic words**	*sound headphones music ear noise cord ipod pair quality head*	*receiver slightly laser pointer green serious blocks kensington delormes beam*
	Mei07	the sound; The sound; sound is	pointer is; of this; laser pointer
	Aletras14	Noise-cancelling headphones; Previous Bose headphones; Bose headphones	Laser safety; Laser pointer; Laser diode
	Top-prob	They have very good sound quality.	I made the mistake to buy the Kensington laser pointer before this one.
	Centroid	I just leave the mounts in place on the windshields of both cars and move the Nuvi and power cord to whichever car I am using at the moment.	I can maybe prefer a bit the ergonomy of the Kensington together with the very practical possibility to store the USB receiver inside the unit but this is probably just my taste and the hiro is quite comfortable either and lighter.
	Sent-label	The output of sound quality from an ipod is amazing.	The Kensington laser pointer is slightly expensive.
IMDb	**Topic words**	*without no want clever intelligent surely play offer effective puts*	*language use vulgar blood used excessive dialog foul us realism*
	Mei07	is one; that it; that there	violence and; of violence; terms of
	Aletras14	Flying Spaghetti Monster; Expelled: No Intelligence Allowed; Prayer	Vampire: The Masquerade ? Bloodlines; Monkeys in Chinese culture; Chavacano
	Top-prob	But what would Lucifer be without his collaborators.	But the story flows well as does the violence.
	Centroid	The Departed is without doubt the best picture of the year and never fails to entertain.	It's far too graphically violent for teenagers, including images of heads being shot and spurting blood, limbs being broken, etc.
	Sent-label	If you want to see an intelligent, memorable thriller, watch The Departed.	The movie includes some overdone foul language and violent scenes.

and Keyphrase against the best performing extractive label (Sent-label) in terms of grammaticality and informativity. In the evaluation, ten raters who are native speakers were presented with a list of top-10 topic words along with one label generated from the following labelling approaches (i.e., Sent-label, PathGraph and Keyphrase) in a random order. For each label, raters were asked to rate the label grammaticality and the label coverage by using the topic words as guidance. In addition, raters were explicitly asked to ignore lack of capitalisation while evaluating grammaticality. The grammaticality of the labels are rated on a 3-points Likert scale:

2 *good sentence* , if the label is a complete grammatical sentence;

1 *almost*, if the label can be understand but requires minor editing, e.g. one mistake in articles;

0 *ungrammatical*, if it is none of the above.

For informativity, labels were rated using the same 4-points

Likert scale as specified in Section 5.1.

Results. Table 7 presents the average grammaticality scores of the sentence labels generated by three different approaches. Unsurprisingly, Sent-label achieves the highest grammaticality score as its labels are the original sentences extracted from the corpus. Although the abstractive methods have lower grammaticality scores, almost 60% of the sentences generated by the Keyphrase algorithm are perfectly grammatical. When comparing the abstractive methods, it is observed that the grammaticality score of Keyphrase is slightly higher than that of PathGraph. This is probably due to the fact that Keyphrase compression generates shorter sentence labels than PathGraph (cf. Table 8). Table 9 shows the human evaluation results for informativity based on Top-1 average rating. We observe a large improvement in informativity for the Keyphrase algorithm, representing an absolute increase of 0.45 over the both Sent-label and PathGraph. It

Table 6: Examples of extractive (Sent-label) and abstractive (PathGraph and Keyphrase) labels for sentiment-bearing topics.

	Positive Topic	**Negative Topic**
Topic words	*pan heat cooking oven stick pans cook stainless food surface*	*money dont product disappointed back waste without very worth make*
Sent-label	I would have opted for the aluminum as that was what I had but my husband bought the stainless steel.	Totally dissatisfy with this product and it's damn waste of money.
PathGraph	i would have opted for the aluminum as that was what i had but my husband bought the stainless steels great too of course but this is a worthy and earthy alternative.	it was so not worth your money!
Keyphrase	it is a good quality stainless and that is important.	very disappointed in product.

Table 7: Human evaluation: Grammaticality ratings, expressed on a scale of 0 to 2. The average ratings (Avg.) are also reported.

System	Grammaticality Rating			
	0	1	2	Avg.
PathGraph	7%	45%	47%	1.40
Keyphrase	7%	35%	57%	1.49
Sent-label	0%	19%	79%	1.80

Table 8: Average label length in words of extractive and abstractive labels across three datasets.

Sent-label	PathGraph	Keyphrase
12.70	13.87	9.90

should also be noted that the average length for Keyphrase is shorter than the average length for both Sent-label and Path-Graph, meaning that Keyphrase can better generate labels with high information coverage of the topic.

Table 6 shows some examples of extractive (i.e., Sent-label) and abstractive (i.e., PathGraph and Keyphrase) labelling for two sentiment-bearing topics extracted from the Kitchen dataset.

To summarise, our experimental results show that the Keyphrase algorithm outperforms the PathGraph algorithm in terms of both grammaticality and informativity. In addition, the abstractive labels generated based on the compression algorithms are more informative than the extractive label, with some cost of reduced grammaticality.

Table 9: Human evaluation: Informativity ratings for extractive and abstractive labels based on Top-1 average rating.

Domain	Sent-label	PathGraph	Keyphrase
Kitchen	2.00	2.10	2.47
Electronics	1.85	1.82	2.10
IMDb	2.14	1.92	2.61
Average	1.90	1.94	2.39

6 Conclusion and Future Work

This paper tackles the problem of automatically labelling sentiment-bearing topics with descriptive sentence labels. We propose two approaches to the problem, one extractive and the other abstractive. Both approaches rely on a novel mechanism to automatically learn the relevance of each sentence in a corpus to sentiment-bearing topics extracted from that corpus. To our knowledge, we are the first to study the problem of labelling sentiment-bearing topics. Our experimental results on three real-world datasets show that both the extractive and abstractive approaches outperform four strong baselines in terms of facilitating topic understanding and interpretation. In addition, when comparing extractive and abstractive labels, abstractive labels are able to provide more topic information coverage despite being shorter, as they can synthesise information from different sentences needed for sentiment-bearing topic interpretations. It should also be noted that our approach does not have any specific dependencies on the JST model, and thus it is general enough to be directly applied to any other sentiment topic model variants which generate multinomial topics as output.

In the future, we would like to extend our work for opinion summarisation. One natural way of achieving this is to summarise documents through the propagation of the document-topic and topic-sentence associations learned from our framework. It is also possible to improve the relevance measure between sentiment-bearing topics and sentences by leveraging external knowledge [38, 39] in addition to textual features.

References

1. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. 3:993–1022, 2003.

2. C. Lin, Y. He, R. Everson, and S. Rüger. Weakly-Supervised Joint Sentiment-Topic Detection from Text. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2011.

3. Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Federica Bisio. Sentic lda: Improving on lda with semantic similarity for aspect-based sentiment analysis. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 4465–4473. IEEE, 2016.

4. Ebuka Ibeke, Chenghua Lin, Adam Wyner, and Mohamad Hardyman Barawi. A unified latent variable model for contrastive opinion mining. *Frontiers of Computer Science*, 2017.

5. J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng. Exploiting topic based twitter sentiment for stock prediction. *ACL*, pages 24–29, 2013.

6. Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 490–499. ACM, 2007.

7. N. Aletras, T. Baldwin, J. Lau, and M. Stevenson. Evaluating topic representations for exploring document collections. *Journal of the Association for Information Science and Technology*, 2015.

8. Yulan He, Chenghua Lin, and Amparo Elizabeth Cano. Online sentiment and topic dynamics tracking over the streaming data. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, pages 258–266. IEEE, 2012.

9. Yulan He, Chenghua Lin, Wei Gao, and Kam-Fai Wong. Dynamic joint sentiment-topic model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):6, 2013.

10. Chengtao Li, Jianwen Zhang, Jian-Tao Sun, and Zheng Chen. Sentiment topic model with decomposed prior. In *SIAM International Conference on Data Mining (SDM 2013). Society for Industrial and Applied Mathematics*, 2013.

11. Jey Han Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. Best topic word selection for topic labelling. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 605–613, 2010.

12. Amparo E. Cano B., Y. He, and R. Xu. Automatic labelling of topic models learned from twitter by summarisation. In *Proceedings of ACL*, 2014.

13. Xian-Ling Mao, Zhao-Yan Ming, Zheng-Jun Zha, Tat-Seng Chua, Hongfei Yan, and Xiaoming Li. Automatic labeling hierarchical topics. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2383–2386. ACM, 2012.

14. J. Lau, K. Grieser, D. Newman, and T. Baldwin. Automatic labelling of topic models. In *Proceedings of ACL*, pages 1536–1545, 2011.

15. Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Un-supervised graph-based topic labelling using dbpedia. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 465–474. ACM, 2013.

16. Nikolaos Aletras and Arpit Mittal. Labeling topics with images using a neural network. In *European Conference on Information Retrieval*, pages 500–505, 2017.

17. Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308, 2014.

18. Md Abul Bashar and Yuefeng Li. Random set to interpret topic models in terms of ontology concepts. In *Australasian Joint Conference on Artificial Intelligence*, pages 237–249, 2017.

19. Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

20. K. Filippova and M. Strube. Sentence fusion via dependency graph compression. In *Proceedings of EMNLP*, pages 177–185, 2008.

21. Micha Elsner and Deepak Santhanam. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 54–63. ACL, 2011.

22. Jackie Chi Kit Cheung and Gerald Penn. Unsupervised sentence enhancement for automatic summarization. In *Proceedings of EMNLP*, pages 775–786, 2014.

23. Kapil Thadani and Kathleen McKeown. Supervised sentence fusion with single-stage inference. In *IJCNLP*, 2013.

24. Hongyan Jing and Kathleen R. McKeown. Cut and paste based text summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 178–185. Association for Computational Linguistics, 2000.

25. Erwin Marsi and Emiel Krahmer. Explorations in sentence fusion. In *In Proceedings of the 10th European Workshop on Natural Language Generation*, pages 109–117, 2005.

26. Emiel Krahmer, Erwin Marsi, and Paul van Pelt. Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In *Proceedings of the 46th Annual Meeting of the ACL on HLT: Short Papers*, pages 193–196, 2008.

27. Ani Nenkova, Kathleen McKeown, et al. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233, 2011.

28. K. Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of COLING*, pages 322–330, 2010.

29. Florian Boudin and Emmanuel Morin. Keyphrase extraction for n-best reranking in multi-sentence compression. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

30. Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1208–1214. AAAI Press, 2015.

31. Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

32. Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

33. John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

34. Thomas P. Minka. Estimating a dirichlet distribution. Technical report, MIT, 2003.

35. Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180, 2003.

36. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354, 2005.

37. Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. Representing topics labels for exploring digital libraries. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 239–248. IEEE Press, 2014.

38. Chenghua Lin, Dong Liu, Wei Pang, and Zhe Wang. Sherlock: A semi-automatic framework for quiz generation using a hybrid semantic similarity measure. *Cognitive computation*, 7(6):667–679, 2015.

39. Rui Mao, Chenghua Lin, and Frank Guerin. Word embedding and wordnet based metaphor identification and interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), 2018.
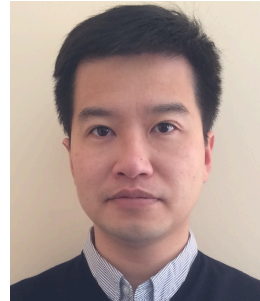
Chenghua Lin received the PhD degree in Computer Science from the University of Exeter in 2011. Currently he is a Senior Lecturer (Associate Professor) in Natural Language Processing at the University of Sheffield. Prior to joining Sheffield in 2019, he held a faculty position at the University of Aberdeen as a SICSA Senior Lecturer. His research focuses on the intersection of Natural Language Processing/Generation and Machine Learning.

Advaith Siddharthan is a Reader in Computational Linguistics at the Open University. He has published over 60 research articles and has worked in Natural Language Processing and Generation, including Text Simplification and Summarisation, for 18 years. Following a Ph.D. at the University of Cambridge, he gained postdoctoral experience at Columbia University and Cambridge before taking up his first faculty position at the University of Aberdeen in 2009 and moving to the Open University in 2017.

Mohamad Hardyman Barawi is currently a PhD student at the University of Aberdeen and a Lecturer at the University Malaysia Sarawak. Prior to the academic post, he was operation support specialist for Hewlett Packard Malaysia. He received his BSc degree (Hons) in Computer Science from University Putra Malaysia in 2003 and an MLIS degree in Library and Information Science from University of Malaya, Malaysia in 2006. His research interests include topic modelling, sentiment analysis, and text summarisation.

Yinbin Liu is an associate professor of the Department of Information Management at the Management School of Shanghai University. He has a PhD in Management Science (Tongji University, China, 2006) and Master degree in Computer Science (Chongqing University, 1999). He has worked in Shanghai University since 2006. He chaired and completed a project supported by the National Social Science Foundation of China. At the same time, he is also a major participant of a project supported by the National Natural Science Foundation of China.