CS474 term project paper

Jihee Park
KAIST
Daejeon, Korea
j31d0@kaist.ac.kr

Junseop Ji
KAIST
Daejeon, Korea
gaon0403@kaist.ac.kr

Soyoung Yoon
KAIST
Daejeon, Korea
lovelife@kaist.ac.kr

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Jihee Park, Junseop Ji, and Soyoung Yoon. 2019. CS474 term project paper. In *CS474 Term project report* . KAIST, ?? pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2 OVERVIEW

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3 TREND ANALYSIS

minted

CS474 Term Project, 2019 Fall, KAIST

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3.1 Data Preprocessing

3.1.1 Data Format. As described in the READ.ME of data provided, The targeted data is from the Korean Herald, National Section news. The period of the dataset is from 2015 to 2017. The Crawled date of the dataset is 2018-10-26. Data format is Json, and there are total of 6 data headers - title, author, time, description, body, and section. Total of 23769 news are included in this dataset.

3.1.2 Load Data. In order to load the data, the instructions recommended at READ.ME are followed. Pandas library is used for better storing and access of the news text.

3.1.3 Libraries Used. For this project, we used pandas and gensim python libraries.

3.2 Previous approaches

Issue trend analysis can be seen as a part of Topic modeling. By searching fields of recent Topic modeling, LDA has shown to have good performance. As a result, LDA is used as a baseline algoritm for this project. A recent study(2018) on Topic Modeling shows that Topic Quality improves when Named Entities are promoted.krasnashchok-jouili-2018-improving This paper proposes 2 techniques: 1.Independent Named Entity Promoting and 2.Document Dependent Named Entity Promoting. Independent Named Entity Promoting promotes the importance of the named entities by applying scalar multiplicaion alpha to the importance of the named entity word. Document Dependent Named Entity Promoting promotes the importance of the named entities by setting the weights of the named entities as maximum term-frequency per document. For Independent Named Entity Promoting, the value of alpha can be changed flexibily, but results conducted by this paper shows that setting alpha as 10 showed the best results. We take advantage of this paper and implement Named Entity Promoted Topic Modeling done by LDA.

3.3 Experiments

3.3.1 Data Tokenization. Lemmatization is not always good At first try, Lemmatization(converting words into base forms) and removal of stopwords were conducted before we run the LDA algorithm and extract Named Entities. We thought that converting words into base forms and reducing the total vocabulary size would increase the performance of topic modeling. Stopwords were taken from , and lemmatization functon was taken from . . But after we do lemmatization, remove stopwords, and tokenize the data, no Named Entities were extracted from the preprocessed corpus. We think the reason for this is as follows. First, words are all converted into lower case when we do lemmatization. This makes the Named Entitiy Recognition system(NER system) to work poorly because we have removed the original information whether the word has a high probability that it is a "Proper pronoun" or not.(). Second,

words are transformed into their base forms, limiting NER system to detect specific words. There also could be cases that the words are transformed into meanings other then their original meanings. For example, "Cooking" and "Cooker" are both converted into "cook" when they are lemmatized, and this makes the word to lose the original information. Third, original relationships between words are lost, because of the removal of stopwords. When we do NER, we have to do the POS tagging of the sentence and then input both the word sequence and the POS sequence of the text. But when we artificially remove stopwords and then do NER, original relationships between words are disrupted and broken. This limits NER system to perform well.

For these 3 reasons, we decided to NOT apply lemmatization for tokenization, because lemmatization lose so much information about the original text and disrupts the NER system's ability to detect Named Entities properly. We decided to just do POS tagging and then do NER. We just used word_tokenize from nltk.tokenize.

3.3.2 Extract NER. By using ne_chunk from nltk and pos_tag from nltk.tag, we extracted Named entities from the original news dataset. NER also extracts multi-word information of Named Entities other than just class ifying whether a word is a named entity or not, so we decided to use that information. We store single-word Named Entities and multi-word named entities separately. As a result, NER and multi-wor d extraction of NER are both processed.

below figure is the topic modeling result(of all time lengths from 2015 to 2017) WITH NER Promoting and WITHOUT NER Promoting. We can see the difference between those two results, and we can conc lude topic modeling with NER promoting shows better performance.

3.3.3 Do LDA. At first try, we ran LDA on naive ner boosted news dataset. but with this approach, we found out that stopwords are classified as top(important)words according to the result of LDA. So we decided to remove stopwords after all the preprocssing(including NER weight promoting)are done. The timing of removal of stopwords are important, as removing stopwords before NER will affect the NER result. Stopword removing are done right before feeding the tokens into LDA. After the removal of stopwords, we could see that the results were much better.

3.3.4 Apply neuroNER. On the topic modeling paper that we referenced says that it uses neuroNER. neuronNER is an easy-to-use program for named entity recognition based on neural networks presented in emnlp 2017. 2017neuroner This neuroNER tool is trained on CONLL2003 dataset and recognizes four types of NE: person, location, organization and miscellaneous. Instead of using , we use to extract Named Entities from the text.

3.3.5 Do LDA with NER promoting. First, split the dataset each year. Then, get tokens for each document with promoted NER frequency (X 10). With this corpus, run the LdaModel with num_topics of 10 and num_words of 30 to 50. Tuning LDA hyperparameters At first, we decided to train the LDA model with num_topics of 10 and num_words of 15. But the results were not very explainable. After experimenting with num_topics and num_words, we found that setting num_topics of 10 is the best representative of the total news. Also, since the only removed word was the stop word, non-ascii character, or unrelated words such as were introduced in the topic result. To extract useful information, we increased num_words for each topics to 50.

4 ON-ISSUE TRACKING

For on-issue tracking, we first divide news articles quarterly. Then we classify news articles in each quarter group into 20 issue categories. For each classified group, each article's 5W1H(when, where, who, what, why, how) is extracted and counted. The most frequent 5W1H will represent an on-issue event for the quarter.

Figure **??** shows the structure of the on-issue tracking process.

4.1 Quarterly Division

We divided all news articles quarterly. The groups contain news articles those are written in *2015 Q1, 2015 Q2, . . . , 2017 Q4, 2018 Q1*. *2018 Q1* group contains only articles written in January, 2018, so we merge the last group with the group *2017 Q4*. The reason why we divided the data quarterly is, the quarter is a semi-standard in the field of yearly statistics. If we divide yearly, there will be only three groups and it will not have high accuracy if we make a timeline of the events. So we chose a quarterly division to make reasonable results.

4.2 Articles in the Quarters Categorization

With LDA model we have trained at trend analysis project, we classify the documents in the quarter groups. If we give a tokenized sentence to the LDA model, the model outputs the probability for each group. We choose the group with maximum value, and assign the document to the group. So, for each quarter, there are 20 classified groups of news articles.

4.3 Event Extraction

For each group we divided from above, we extract the events with the approach of word frequency. For this step, we use a Python library called "giveme5W1H". The library is the state-of-the-art tool for extracting *when/where/who/what/why/how* features from the document. The library uses Stanford's CoreNLP library as its basic structure, and give analysis results when we give a title, lead, text, and a published date. We decided to use columns *title*, *description*, *body*, and a *time* from the given dataset as an input to get a result. For each group, we count the frequencies of each feature of the articles, and select the most frequent terms for each feature, treat them as an event.

5 OFF-ISSUE TRACKING

For off-issue tracking, we first categorize topics given as Trend analysis part. In this section, we denote a document as sequence of tokens plus its created time $\mathbb{D} := (\Sigma^+, t)$, when $t \in \mathbb{R}$ (timestamp of creation time). and the set of document of topic a as $\mathbb{T}_a \in \mathcal{P}(\mathbb{D})$.

5.1 BoW extractuion

In first, we have to extract document in some space which we can analyze quantatively. We use BoW as morphism from document space to vector space $\mathbb{R}^N$, which we can analyze similarity of document. In addition, we add one more dimension to give information of document creation time. From pre-calculated set of

tokens $\Sigma := \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, our transformation $b : \mathbb{D} \to \mathbb{R}^{n+1}$ is defined inductively as

$$\begin{cases} b([], t) := t * e_{n+1} \\ b(\sigma_i :: tl, t) := e_i + b(tl, t) \end{cases}$$

Then, morphism from $\mathbb{T}_a \in \mathcal{P}(\mathbb{D})$ to $\mathcal{P}(\mathbb{R}^{n+1})$ is naturally induced from b as $\phi(\mathbb{T}_a) = \{b(d) | d \in \mathbb{T}_a\}$

5.2 Relation between semantic of document and BoW

We know that there are documents and events which have similar meaning, but we cannot formalize it because we currently do not have model of language interpretation in metric space. But we can assume *such* space exists, i.e. there is an isomorphism $\phi : \mathbb{D} \to \mathbb{D}^\#$, when $(D^\#, d^\#)$ is metric space. It is not hard to assume this structure, since similar concept is already introduced as Entity comparison/Behavior comparison operator of Semantic algebra [**?**].

Our desired result is that b with euclidean distance successfuly models $(D^\#, d^\#)$, but we cannot show it because we do not have constructive definition of $D^\#$. But if it has sufficient approximation, (bounded approximation) We can derive more interesting properties (such as bounded error from BoW to Event space, etc).

Definition 5.1. b has approximation of ϕ with bound K, ϵ iff there exists an Lipshitz continuous π with K that $d^\#(\pi(b(d)), \phi(d)) \leq \epsilon$.

5.3 Relation between semantic of event and BoW

Once semantic of document is defined, we can build similar notion of event as metric space. To build such space, we first understand about relation between document and event.

- similar document refer similar event.
- similar event (even same event) may be refered by documents with far distance, but it is not arbitrarly far.

we can formulize this as logical formlua, with definition of $e : D^\# \to E^\#$. $((E^\#, e^\#)$ is metric space for event)

- if $d^\#(d_1, d_2)$ is sufficiently small, then $e^\#(e(d_1), e(d_2))$ is sufficiently small.
- when $e^\#(e(d_1), e(d_2))$ is small, it doesn't mean $d^\#(d_1, d_2)$ is small but is bounded.

begin with this fact, we can find very interesting property which generalize this: continuity.

Definition 5.2. e is Lipschitz continuous with K if and only if $e^\#(e(d_1), e(d_2)) \leq K d^\#(d_1, d_2)$.

We can check that if e is Lipschitz continuous with K_e, then above two property is satisfied. Also, it derives important fact: If we have approximation of semantics with bounded error, then there also exists approximation of event with bounded error.

Theorem 5.3. *b has approximation of ϕ with bound K, ϵ, then there exists $\pi_e : \mathbb{R}^{n+1} \to E^\#$ s.t. $e^\#(\pi_e(b(d)), e(\phi(d))) \leq K_e \cdot \epsilon$. (it means b has approximation of $e \cdot \phi$ with bound $K, K_e \cdot \epsilon$)*

Although proof is directly derived from Lipschitz continuity, it emphasizes that if we have bounded approximation of document, then it guarantees bounded approximation of event.

5.4 Event clustering

In this assumption about semantic of document ans event, we can build event clustering method. Before using techniques in R^{n+1}, we focus on how this clustering in R^{n+1} effects in $E^\#$.

Theorem 5.4. *if b has approximation of $e \cdot \phi$ with bound K, ϵ, then $e^\#(e \cdot \phi(d_1), e \cdot \phi(d_2)) \leq 2 \cdot \epsilon + K||b(d_1) - b(d_2)||$.*

Proof.

$$\begin{aligned} & e^\#(e \cdot \phi(d_1), e \cdot \phi(d_2)) \leq e^\#(e \cdot \phi(d_1), \pi_e(b(d_1))) + \\ & e^\#(\pi_e(b(d_1)), \pi_e(b(d_2))) + e^\#(\pi_e(b(d_2)), e \cdot \phi(d_2)) \leq \\ & \epsilon + e^\#(\pi_e(b(d_1)), \pi_e(b(d_2))) + \epsilon \leq \\ & 2 \cdot \epsilon + K||b(d_1) - b(d_2)||. \end{aligned}$$

□

It shows that, if we make good Vector transformation b, then it automatically guarantees bounded error for distance of extracted event, without construction of π, ϕ, e or any other. Begin with this fact, we derive constructive definition of partition for documents using approximated transformation b. To do that, we first define similarity relation for two documents.

Definition 5.5 (Similarity relation). $\approx_{\mathbb{R}^{n+1}, \delta} \in \mathcal{P}(\mathbb{D} \times \mathbb{D})$ is defined as

$$d_1 \approx_{\mathbb{R}^{n+1}, \delta} d_2 \iff ||b(d_1) - b(d_2)|| \leq \delta.$$

Similarly, $\approx_{E^\#, \delta} \in \mathcal{P}(\mathbb{D} \times \mathbb{D})$ is defined as

$$d_1 \approx_{E^\#, \delta} d_2 \iff e^\#(e \cdot \phi(d_1), e \cdot \phi(d_2)) \leq \delta.$$

then $\approx_{\mathbb{R}^{n+1}, \delta} \subseteq \approx_{E^\#, 2 \cdot \epsilon + K \cdot \delta}$ holds by above theorem. Thus it is quite reasonable to use $\approx_{\mathbb{R}^{n+1}, \delta}$ to cluster events, instead of uncomputable relation $\approx_{E^\#, 2 \cdot \epsilon + K \cdot \delta}$.

Definition 5.6 (Transitive closure). $\approx^*_{\mathbb{R}^{n+1}, \delta}$ is smallest relation on $\mathbb{D}$ that contains $\approx_{\mathbb{R}^{n+1}, \delta}$ and is transitive.

Then $\approx^*_{\mathbb{R}^{n+1}, \delta}$ is reflexive, symmetric and transitive, which can be considered as equivalence relation. Then, we can partition documents with this equivalence relation.

Definition 5.7 (Partiton of $\mathbb{D}$). when $\approx$ is equivalence relation, $\mathbb{D}/\approx := \{[a] | a \in \mathbb{D}\}$, when $[a] := \{b \in \mathbb{D} | a \approx b\}$.

By substitute $\mathbb{D}$ to $\mathbb{T}_a$, finally we have $\mathbb{T}_a / \approx^*_{\mathbb{R}^{n+1}, \delta}$ as successful approximation of event partition of topic a. Now, we are going to explain how most relevent description of event is extracted from each partiton.

5.5 Extracting representative description

Now we have cluster of events (documents which describing events) $\mathbb{T}_a / \approx^*_{\mathbb{R}^{n+1}, \delta}$, but we should return summary of events, because whole collection of documents are quite long to read and might have unnecessary information. So we have to extract *representative description* of tht event cluster. To extract target information from a document is well studied in information extraction field, and there

are several method such as template-based information extraction, neural methods, etc. But in the case of several docuemnts, it is hard to converge summary to cover all document's information, because existing works is not based on language semantic-based, so it is hard to generate summary statement bewtween description of similar/same meaning.

For example, if one document describe the event happens "one day after of 12/7", and there are another document describe the event was happend "one day before of 12/9". Obviously both description refer same day, but token-based approach (or pattern-based approach such as signal words) cannot handle this issue. Even with this disadvantage, above method is widely used because of its high performance (and due to challenges of semantic based information extraction method).

So, we decided to use event extractor for one document, but we design to choose representative document appropriately.

Definition 5.8 (Representative docuemnt). document $d \in [a]$ is *representative document* of $[a]$ when $\sum_{d' \in [a]} ||b(d) - b(d')||$ $\leq \sum_{d' \in [a]} ||b(x) - b(d')||$ for any $x \in [a]$.

It means that we choose to extract event from a document which has minimum difference between all other documents. After choosing representative document, we use Giveme5W1H framework[**?**] to extract description of event.

5.6 implementation

To implement BoW transformation and document clustering, we use pandas and gensim for python. to calculate transitive closure and finding partition, we use DBSCAN algorithm. Parameters are adjusted by experiments on small set of documents. After that, extracting event description is done by Giveme5W1H framework.

6 EVALUATION

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

7 CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

afterner.png

beforener.png

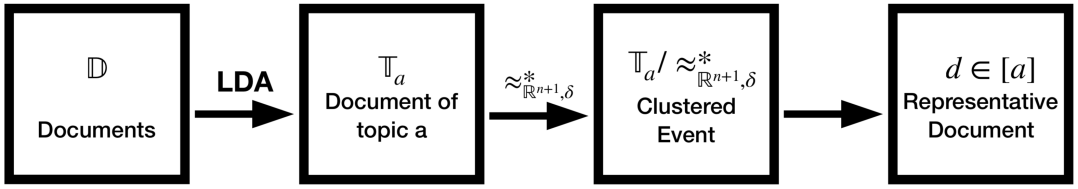

Figure 2: Overview of off-issue tracking process.