Integrating Latent Dirlichet Allocation to Centroid-based Text Summarization

Hair A. Parra B.
McGill University
jair.parra@outlook.com

Logan Ralston
McGill University
logan.ralston@mail.mcgill.ca

James Berry
McGill University
james.berry@mail.mcgill.ca

Abstract

In this paper, we adapt a recent centroid-based text summarization model, one that takes advantage of the compositionality of word embeddings, in order to obtain a single vector representation of the most meaningful words in a given text. We propose utilizing Latent Dirichlet Allocation (LDA), a probabilistic generative model for collections of discrete data, in order to better obtain the topic words of a document for use in constructing the centroid vector. We see that the LDA implementation results in overall more coherent summaries, suggesting the potential for utilizing topic models to improve upon the general centroid-based method.

1 Introduction

Automatic text summarization is the task of producing a text summary "from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that" (Dragomir R Radev and McKeown, 2002). Established work in the area of text summarization can be divided generally into two approaches: extractive summarization, which builds summaries using selected sentences from the original text itself; and abstractive summarization, which aims to generate original sentences containing the key information from the original text (Mehdi Allahyari and Kochut, 2017). The majority of work in the literature is currently extractive-focused, due to extractive-based methods avoiding the difficulties of emulating human language capabilities present in natural language generation - and, in fact, as of present existing abstractive summarizers often rely on an extractive preprocessing step before generating a summary based off of the extracted sentences (Mehdi Allahyari and Kochut, 2017).

Recent work by (Gaetano Rossiello and Semeraro, 2017) proposes a centroid-based extractive model that makes use of the inherent compositionality of word embeddings to easily encode the notion of semantic similarity between words. Despite being a relatively computationally simple model, it performs well even compared to more complex deep-learning models (Gaetano Rossiello and Semeraro, 2017). The centroid-based method constructs a centroid vector, consisting of a sum of the embeddings of the words considered most relevant within the text, and constructs the summary from the sentences most similar to the centroid vector. This concept of a centroid vector was introduced by (Dragomir R. Radev and Tam, 2004), but utilizing a bag-of-words (BOW) model. By instead utilizing a word embedding model Rossiello et al. were able to encode the notion of similarity between words, allowing the resulting summarization model to to extract sentences similar to the centroid vector even if they contained none of the words in the centroid itself.

In order to construct the centroid vector, Rossiello et al. utilize the *tf*idf* scheme (Robertson, 2013), summing the word vectors with *tf*idf* weight above a given threshold. We propose instead utilizing Latent Dirichlet Allocation (David M. Blei and Jordan, 2003) as a topic model, from which we can extract the most relevant words to utilize for the centroid vector. Due to the notion of similarity encoded in the word embeddings outlined above, we are not restricted to using words in the text to be summarized for the centroid vector, and so we hypothesize that our model will result in improved summaries compared to the original *tf*idf* model, due to the centroid vector

being composed from the most relevant words to a topic from the entire corpus, not just the words present in the text to be summarized.

In the section below, we briefly outline word embeddings and some current text summarization methods in the literature, as well as further elaborate on the centroid-based technique and LDA.

2 Related Work

2.1 Text Summarization and Word Embeddings

Initial work in the literature has often utilized BOW models, which consist of creating vector representations via a one-hot encoding. As stated above, the limitation to these models is the inability to encode a notion of similarity between word representations, and as such there has been wide efforts to propose alternative methods that utilize non-BOW representations of words. These alternatives include matrix factorization methods like Latent Semantic Analysis (LSA) (Makbule G. Ozsoy and Cicekli, 2011), as well as Deep Learning models that have proven extremely successful in recent years, such as (Ziqiang Cao and Zhou, 2015). LDA-based models have also proven popular, for example BayeSum, a Bayesian text summarization model based off of LDA (Daumé and Marcu, 2006). An excellent survey of some popular approaches to text summarization has been written by (Mehdi Allahyari and Kochut, 2017)

Word embedding refers to a method of constructing continuous vector representations of words in text, able to encode both information about the word itself as well as in relation to other words (i.e their similarity). One of the most famous models for word embeddings is Word2vec(Tomas Mikolov and Dean, 2013), which makes use of a a single-layered neural network model to learn complex relationships between words, encoding them as continuous-valued vectors with meaningful vector substructure(Tomas Mikolov and Dean, 2013)(Gaetano Rossiello and Semeraro, 2017). Word2Vec supports both continuous bag-of-words (CBOW) and skip-gram representations of word vectors. As stated in Rossiello et al. paper, the centroid-based model is general, permitting the usage of either of these representations (or in fact any other word embedding model) (Gaetano Rossiello and Semeraro, 2017). In accordance with their paper, we train both a CBOW and skip-gram based model. In the following section, we briefly describe how these components make an essential component of the summarization technique proposed in *Centroid-base Text Summarization through Compositionality of Word Embeddings*(Gaetano Rossiello and Semeraro, 2017).

2.2 Centroid-Based Summarization

In order to explore textual similarity, a crucial aspect for many extractive text summarization methods Rossiello et al. propose a "centroid-based method for text summarization that exploits the compositional capabilities of word embeddings"(Gaetano Rossiello and Semeraro, 2017). The model intention is simple: create document centroid-embeddings that represent the most important, or "central" idea of the given paragraph, so that sentences that are most similar to this centroid are selected for text summarization. Formally, they define their model is described as follows(Gaetano Rossiello and Semeraro, 2017):

1. Given a corpus of documents $[D_1, D_2, \dots]$ and its vocabulary V with size $N = |V|$,define a matrix $E \in \mathbb{R}^{N,k}$, so-called *lookup table*, where the i-th row is a word embedding of size $k < N$.of the i-th word in V.

2. The document is split into sentences, words are converted into lowercase and stopwords are removed.

3. The **Centroid embedding** for a document is created by the summation of those words in the given document that have a $tf * idf$(Robertson, 2013) weight greater than a threshold t.

$$C = \sum_{w \in D, tfidf(w) > t} E[idx(w)] \quad (1)$$

where $idx(w)$ returns the index of the word in the lookup table described above.

4. **Sentence embeddings** are created by summing all the word embeddings in that sentence, i.e.

$$S_j = \sum_{w \in S_i} E[idx(w)] \quad (2)$$

5. Calculate similarity between centroid C and each sentence S_j in the document using cosine similarity:

$$sim(C, S_j) = \frac{\langle C, S_j \rangle}{||C||\,||S_j||} = \frac{C^T \cdot S_j}{||C||\,||S_j||} \tag{3}$$

6. Sort sentences in descending order of their similarity scores. The top ranked sentences are iteratively selected and added to the summary until a limit(in bytes or words) is reached, while also computing cosine similarity between sentences and ignoring sentences that are too similar to avoid redundancy.

In the next section, we describe the LDA model, and in section 4 we describe how we can integrate this to improve the Centroid-based summrization just described.

2.3 Latent Dirichlet Allocation

Latent Dirlichet Allocation is a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model in which each item of a collection if modeled as an infinite mixture over and underlying set of topic probabilities. (David M. Blei and Jordan, 2003). In particular, Ng et al. present in their paper "efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes estimation". The model is described as follows:

1. A *word* is a discrete item represented by a one-hot encoded vector $w \in \mathbb{R}^V$ from a vocabulary $\{1, \ldots, V\}$ such that $w^i = 1$, whenever w represents the i-th word in the vocabulary and zero otherwise.

2. A *document* is a sequence of N words , denoted by $\mathbf{w} = (w_1, \ldots, w_N)$,where w_n is the $n - th$ word in the sequence.

3. A *corpus* is a collection of M documents denoted by $\mathcal{D} = \{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$

The LDA then assumes the following generative model:

1. Choose $N \sim$ Poisson(ξ)

2. Choose $\theta \sim$ Dir(α) (The Dirlichet Distribution)

3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim$ Multinomial(θ)
   (b) Choose a word w_n from $p(w_n|z_n, \beta)$, where $w_n|z_n \sim$ Multinomial(β).

In this case, initially assuming a fixed dimensionality for the Dirlichet distribution, we have $\beta \in \mathbb{R}^{kxV}$,where $\beta_{ij} = p(w^j = 1|z_i = 1)$ or in simple words, the probability that we chose the word j given the topic i. The Dirlichet random variable θ with support on the $(k-1)$-simplex (i.e. $\theta \in \mathbb{R}^k$ s.t. $\theta_i \geq 0, \sum_{i=1}^k = 1$) has the probability density

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_i^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha^i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1}$$

where $\alpha \in \mathbb{R}^k, \alpha^i \geq 0$ and $\Gamma(.)$ is the gamma distribution. Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is given by:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$$

where $p(z_n|\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over z, the **marginal distribution of a document** is given by:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta$$

Finally, taking the product of the marginal probabilities of single documents, the **probability of the corpus** is given by:

$$p(\mathcal{D}|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d$$

The LDA model (David M. Blei and Jordan, 2003), provides us with a technique that allows us to model complex probabilistic relationships between the different levels that compose a given corpus (i.e. words, documents, "topics" and the corpus itself). We will see in section 4 how we integrate this approach to further enhance the Centroid-based Model.

3 Dataset and setup

We make use of data from the 2004 Document Understanding Conference (DUC) (Chali and Kolla, 2004) in order to most effectively compare our model with that of (Gaetano Rossiello and Semeraro, 2017), who used the same dataset (The DUC-2004 Task 2 dataset) to train and evaluate their model. The dataset consists of press releases from the Associated Press and New York Times newspapers, clustered in 50 clusters of 10 documents each. To evaluate our model we make use of ROUGE, a set of metrics for evaluating text summarizations against some gold standard, based on n-gram overlap (Lin, 2004). For concurrency with the original paper, we also utilize ROUGE-1 and ROUGE-2 as scoring metrics. In addition to comparison with Rossiello et al.'s models, we also include the winning model from DUC-2004 Task 2, **Peer65**. In addition as baselines, we include scores for - **SumBasic** (Nenkova and Vanderwende, 2005), a simple probabilistic model often used as a lower bound summarization model; and as an upper bound we include the recurrent neural network model proposed by (Ziqiang Cao and Zhou, 2015), listed as **RNN**. (ROUGE scores for these models all taken from (Gaetano Rossiello and Semeraro, 2017))

4 Methodology and Implementation

We base our work mostly on Rossiello et al.'s (Gaetano Rossiello and Semeraro, 2017) summarizing technique described in section 2.2. We propose a simple change on this model based on the ideas described in section 2.3: instead of composing words that have a $tf * idf$ score higher than a threshold, we can instead model the document as an underlying mixture of topics which in turn are represented as a linear combination of representative words with different densities. The process is described as follows:

1. Once the LDA algorithm is run on the corpus, we'll obtain as an output l topics with N_l words each (David M. Blei and Jordan, 2003). Each topic T_l, is turn are represented as a linear combination of words and densities that compose them, that is:

$$T_l = \sum_{w \in T_l} w * p(w|T_l) \qquad (4)$$

2. Given a new document D_{new}, the LDA model will return the most likely topics for it, represented as mixtures given in equation above (4).

3. We denote by $LDA(D)$:=$LDA(D, m, n_m)$ the set of all words that belong to the n_m most likely words for each of the m most likely topics for the document, as assigned by the LDA.

4. Finally, we construct the LDA centroids as

$$C = \sum_{w \in LDA(D)} E[idx(w)] \qquad (5)$$

We use the resulting LDA-centroids to run their algorithm in equation 5 instead of those in equation 1. Since the LDA algorithm performance is affected by the quality of the training data text, we implement our own `preprocessor` mini-module for text cleaning, including classes based on either `nltk`[1] or `spaCy`[2], since the latter also has more specialized tools to work with other languages such as French or Spanish. Although we do not test these in this project, the implementations are implemented keeping these language extensions to the summarization algorithm as well. The specific preprocessing steps we apply to train the LDA algorithm include removing stopwords, punctuation, numbers, symbols and determiners, since these do not belong to open classes and therefore do not contribute significantly to topic formation. In order to obtain the word embeddings, we use the `word2vec` from the `gensim` library with the same training parameters as Rossiello et al. (Gaetano Rossiello and Semeraro, 2017) to train the CBOW and Skip-Gram models. As with Rossiello et al., we do not perform any kind of stemming, allowing the word embeddings to learn the similarities between words with the same roots (Gaetano Rossiello and Semeraro, 2017). In training our model, the major hyperparameter are the number of topics learned by the LDA model, and the number of words from the topic chosen for the centroid vector. To find the best hyperparameter configuration, we perform grid search with values of [32, 50, 64, 100], [20,30,40,50] for the respective parameters outlined above, resulting in values of 64, 20, respectively.

[1] https://www.nltk.org/
[2] https://spacy.io/

5 Results

The results of the experiment are shown in Table 1. We report the best scores of our models as **C_LDA_C** and **C_LDA_S** for the CBOW and skip-gram models respectively, with the original models as **C_CBOW** and **C_SKIP** respectively, as well as the baseline comparisons listed in Section 3. For each metric, the best result is listed in bold.

Both of our models beat the lower bound for both scores, and outperform all models (including the RNN based model) on the ROUGE-2 metric. With regards to the original centroid paper our model returns similar, albeit slightly worse ROUGE-1 scores, but better ROUGE-2 scores. This shows the validity of using LDA to generate centroid vectors, and suggests that our model generates more coherent summaries, at the expense of slightly less coverage.

Model	*ROUGE-1*	*ROUGE-2*
SumBasic	37.27	8.58
Peer65	38.22	9.18
RNN	38.78	9.86
C_CBOW	38.68	8.93
C_SKIP	**38.81**	9.97
C_LDA_C	37.53	10.18
C_LDA_S	37.93	**10.24**

Table 1: ROUGE Scores (%) on the DUC-2004 Task 2 Dataset

6 Discussion and Conclusion

In this paper we propose a refinement of the centroid-based summarization model of (Gaetano Rossiello and Semeraro, 2017), in the form of utilizing LDA to construct centroid vectors. As seen in the results, our model generates comparative summaries in terms of coverage, but with an improvement in terms of summary coherence. This shows the value of a more fine-grained construction of the centroid, resulting in more coherent, human-like summaries. In addition, our model shows the potential for combining topic modelling methods with the centroid-based model - the centroid model itself is clearly general, and our model shows that it has potential to be implemented with a variety of centroid construction methods.

We see the possibility of further improving our model, via a possibly more fine-grained approach to extracting meaningful words from the LDA model. For example we see potential in extracting words based off of a probability threshold, or from training multiple LDA models on more focused datasets (for example the individual clusters of the DUC dataset) then for a given document identifying the specific model it best fits, then constructing the centroid from that specific LDA model. As well, as stated above the centroid model seems to be extendable for use with any kind of topic model, and we are interested with testing implementation with other topic models such as Latent Semantic Analysis (Makbule G. Ozsoy and Cicekli, 2011), or Non-Negative Matrix Factorization (Michael W. Berry and Plemmons, 2007).

7 Statement of Contributions

Hair Parra : Github Repository setup, project design planning; LDA implementation, testing; final report related work, methodology, conclusion.

Logan Ralston: Project design planning; centroid implementation, testing; final report results, conclusion.

James Berry: Project design planning; rouge implementation, testing; source research, final report abstract, intro, related work, dataset, conclusion.

8 Source Code

The code used for this project can be found publically at `https://github.com/JairParra/LDA_centroid_based_summarization`.
NOTE: The testing datasets must be requested from the DUC.

References

Yllias Chali and Maheedhar Kolla. 2004. Summarization techniques at duc 2004. *online: https://arxiv.org/pdf/1301.3781.pdf*.

Hal Daumé and Daniel Marcu. 2006. Bayesian query-focused summarization. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, page 305–312.

Andrew Y. Ng David M. Blei and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Eduard Hovy Dragomir R Radev and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational Linguistics*, 28:399–408.

Malgorzata Stys Dragomir R. Radev, Hongyan Jing and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing Management*, 40:919–938.

Pierpaolo Basile Gaetano Rossiello and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. *online:* https://www.aclweb.org/anthology/W17-1003.pdf.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. pages 74–81.

Ferda N. Alpaslan Makbule G. Ozsoy and Ilyas Cicekli. 2011. Text summarization using latent semantic analysis. *Journal of Information Science*, 37:405–417.

Mehdi Assefi Saeid Safaei Elizabeth D. Trippe Juan B. Gutierrez Mehdi Allahyari, Seyedamin Pouriyeh and Krys Kochut. 2017. Text summarization techniques: A brief survey. *online:* https://arxiv.org/pdf/1707.02268.pdf.

Amy N. Langville V. Paul Pauca Michael W. Berry, Murray Browne and Robert J. Plemmons. 2007. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics Data Analysis*, 52:155–173.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. *online=https://www.cs.bgu.ac.il/ elhadad/nlp09/sumbasic.pdf.*

Stephen Robertson. 2013. Understanding inverse document frequency: On theoretical arguments for idf. *online:* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.438.2284&rep=rep1&type=pdf.

Greg Corrado Tomas Mikolov, Kai Chen and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *online:* https://arxiv.org/pdf/1301.3781.pdf.

Li Dong Sujian Li Ziqiang Cao, Furu Wei and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. *Proceedings of the Twenty Ninth AAAI Conference on Artificial Intelligence*, page 2153–2159.