A study of semantic augmentation of word embeddings for extractive summarization

Nikiforos Pittaras
DIT, NKUA
IIT, NCSR-D
npittaras@di.uoa.gr

Vangelis Karkaletsis
IIT, NCSR-D
vangelis@iit.demokritos.gr

Abstract

In this study we examine the effect of semantic augmentation approaches on extractive text summarization. Wordnet hypernym relations are used to extract term-frequency concept information, subsequently concatenated to sentence-level representations produced by aggregated deep neural word embeddings. Multiple dimensionality reduction techniques and combination strategies are examined via feature transformation and clustering methods. An experimental evaluation on the MultiLing 2015 MSS dataset illustrates that semantic information can introduce benefits to the extractive summarization process in terms of F1, ROUGE-1 and ROUGE-2 scores, with LSA-based post-processing introducing the largest improvements.

1 Introduction

In recent years, the abundance of textual information resulting from the proliferation of the Internet, online journalism and personal blogging platforms has led to the need for automatic summarization tools. These solutions can aid users to navigate the saturated information marketplace efficiently via the production of digestible summaries that retain the core content of the original text (Yogan et al., 2016). At the same time, advancements introduced by deep learning techniques have provided efficient representation methods for text, mainly via the development of dense, low-dimensional vector representations for words and sentences (LeCun et al., 2015). Additionally, semantic information sources have been compiled by humans in a structured manner and are available for use towards aiding a variety of natural language processing applications. As a result, semantic augmentation approaches can introduce existing knowledge to the neural pipeline, circumventing the need for the neural model to learn all useful information from scratch.

In this study, we examine the effect of semantic augmentation and post-processing techniques on extractive summarization performance. Specifically, we modify the input features of a deep neural classification model by injecting semantic features, simultaneously employing feature transformation post-processing methods towards dimensionality reduction and discrimination optimization. Specifically, we aim to address the following research questions.

- Can the introduction of semantic information in the network input improve extractive summarization performance?
- Does the semantic augmentation process benefit via dimensionality reduction post-processing methods?

The rest of the paper is structure as follows. In section 2 we cover existing related work relevant to this study. This is followed by a description of our approach (section 3). In section 4 we outline our experimental methodology and discuss on results and findings. Finally, we present our conclusions in section 5.

2 Related work

2.1 Text representations

Extensive research has investigated methods of representing text for Natural Language Processing and Machine Learning tasks.

Vector Space Model (VSM) approaches project the input to a n-dimensional vector representation, exploiting properties of vector spaces and lin-

Proceedings of the Multiling 2019 Workshop, co-located with the RANLP 2019 conference, pages 63–72
Varna, Bulgaria, September 6, 2019.
http://doi.org/10.26615/978-954-452-058-8_009

ear algebra techniques for cross-document operations. Approaches like the Bag-of-Words (Salton et al., 1975) have become popular baselines, mapping the occurence of an input term (e.g. a word) to their occurence frequencies in the text. Modifications to the model include refinements in the term weighting strategy such as DF and TF-IDF normalizations (Yang, 1997; Salton and Buckley, 1988), term preprocessing such as stemming and lemmatization (Jivani et al., 2011), and others. Further, sentence and phrase-level terms are examined (Scott and Matwin, 1999), along with n-gram approaches, which consider n-tuple occurences of terms instead (Brown et al., 1992; Katz, 2003; Post and Bergsma, 2013).

Other approaches encode term co-occurence information via representation learning, relying on the distributional hypothesis (Harris, 1954) to capture semantic content. At the same time, the need to circumvent the curse of dimensionality (Hastie et al., 2005) of term-weight feature vectors has led to the production dense, rather than sparse representations. Early such examples used analytic matrix decompositions on co-occurence statistics (Jolliffe, 2011; Deerwester et al., 1990; Horn and Johnson, 2012), while more recently, vector embeddings are iteratively optimized learned by analyzing large text corpora using local word context in a sliding window fashion (Mikolov et al., 2013a,b), or using pre-computed pairwise word co-occurences (Pennington et al., 2014). More refined methods break down words to subword units (Bojanowski et al., 2017), where learning representations for the latter enables some success in handling out-of-vocabulary words.

2.2 Extractive summarization

In summarization, contrary to the abstractive approach where output summaries are generated from scratch (Yogan et al., 2016), the extractive method relies on sentence salience detection to retain a minimal subset of the most informative sentences in the original text (Gupta and Lehal, 2010). VSM approaches have been widely utilized in sentence modelling for this task, with a variety of methods for determining term weights based on word frequency, probability, mutual information or tf-idf features, sentence similarity, as well as a variety of feature combination methods (Mori, 2002; McCargar, 2004; Nenkova and Vanderwende, 2005; Galley et al., 2006; Lloret and Palomar, 2009). Other popular handcrafted features used are syntactic / grammar information such as part-of-speech tags, as well as sentence-wise features such as sentence position and length. Finally, similarity scores to title, centroid clusters and predefined keywords can be used to score / rank sentences towards salience identification and extraction (Neto et al., 2002; Yogan et al., 2016).

Other works adopt a topic-based approach, using topic modelling techniques towards sentence salience detection. For example, the work in (Aries et al., 2015) builds topics via a clustering process, using a word and sentence-level vector space model and the cosine similarity measure. Clustering techniques have been applied to this end, for sentence grouping and subsequent salience identification (Radev et al., 2000).

Graph methods have also been exploited; In (Lawrie et al., 2001), the authors adopt a graph-based probabilistic language model towards building a topic hierarchy for predicting representative vocabulary terms. The MUSE system (Litvak and Last, 2013) combines graph-modelling with genetic algorithms towards sentence modelling and subsequent ranking, while the work in (Mihalcea and Tarau, 2004) builds sentence graphs using a variety of feature bags and similarity measures and proceeds to extract central sentences via multiple iterations of the TextRank algorithm.

2.3 Semantic enrichment

Semantic information has been broadly exploited towards aiding NLP tasks, using resources such as Wordnet (Miller, 1995), Freebase (Bollacker et al., 2008), Framenet (Baker et al., 1998) and others. Such external knowledge bases have seen widespread use, ranging from early works on expanson of rule-based discrimination techniques (Scott and Matwin, 1998), to synonym-based feature extraction (Rodriguez et al., 2000) and large-scale feature generation from WordNet synset relationships edges for SVM classification (Mansuy and Hilderman, 2006).

In extractive summarization, semantic information has been used as a refinement step in the sentence salience detection pipeline. For example, in (Dang and Luo, 2008), the authors utilize WordNet synsets as a keyphrase ranking mechanism, based on candidate synset relevance to the text. Other approaches (Vicente et al., 2015) use semantic features from Wordnet and named entity extrac-

tion, followed by a PCA-based post-processing step for dimensionality reduction. Wordnet is also utilized in (Li et al., 2017) where the authors use the resource for sentence similarity extraction, using synset similarity on the word level and treating the resulting scores as additional features for summarization and citation linkage.

Our approach bears some similarities with the work of (Vicente et al., 2015), extending the investigation to additional post-processing techniques to PCA, examining post-processing application strategies, and adopting deep neural word embeddings as the lexical representation, while grounding on a number of baselines. In the following section, we will describe our approach in detail, including text representation, semantic feature extraction, training and evaluation.

3 Proposed Method

3.1 Problem definition

We formulate the task of automatic summarization as a classification problem. Given a document consisting of N sentences $D = \{s_1, s_2, \dots s_N\}$ and a ground truth (extractive) summary of size k, $G = \{g_1, g_2, \dots g_k\}$, $g_i \in D$, a classification-based extractive summarization system F selects salient sentences $P = \{p_1, p_2, \dots p_k\}$ via $F(D) = P$, such that P is as close to G as possible. In this work, $F(\cdot)$ is a data-driven machine learning model, built by exploiting statistical features in the input text.

3.2 Text representation

We use a variety of approaches for representing the textual component of a sentence. First, we employ Continuous Bag-of-Words (CBOW) variant of the popular word2vec model (Mikolov et al., 2013b), which builds vector representations of a word using a statistical language model that predicts the word based on its surrounding context. More formally, given a center word in a sentence, w_c and and a set of $2k$ context words around it $w_{context} = [w_{c-k} \dots, w_{c-1}, w_{c+1}, \dots, w_{c+k}]$, CBOW tries to optimize the conditional probabilistic neural language model $P(w_c|w_{context})$.

We train embeddings from scratch with this method, optimizing with the cross-entropy loss, ending up with a vector representation for each word in the dataset. We subsequently produce the final, sentence-level representation by averaging the vectors corresponding to words in a sentence. In addition to embedding training, we examine the performance of pre-trained Fastext (Joulin et al., 2016) embeddings, produced by a model that captures subword information via character embeddings, enabling handling of out-of-vocabulary words. Additionally, we employ direct sentence-level modelling alternatives via the doc2vec (Le and Mikolov, 2014) extension of word2vec, as well as a sentence-level TF-IDF baseline.

3.3 Semantic representation

In order to capture and utilize semantic information in the text, we use the WordNet semantic graph (Miller, 1995), a lexical database for English, often used as an external information source for machine learning research in classification, summarization, clustering and other tasks (Hung and Wermter, 2004; Elberrichi et al., 2008; Liu et al., 2007; Morin and Bengio, 2005; Bellare et al., 2004; Dang and Luo, 2008; Pal and Saha, 2014). In Wordnet, semantic relations between concepts are captured in a semantic graph of synonymous sets (*synsets*), as well as multiple types of relations of lexical / semantic nature, such as ike hypernymy and hyponymy (is-a graph edges), meronymy (part-of relations, and others). We employ WordNet as an enrichment mechanism, extracting frequency-based features from corpus words. Specifically, we mine semantic concepts from each word in the text, arriving at a sparse high-dimensional bag-of-concepts for each document. This vector is concatenated to the lexical representation. To deal with the curse of dimensionality (Hastie et al., 2005) of this approach, we apply dimensionality reduction via PCA (Jolliffe, 2011), LSA (Deerwester et al., 1990) or K-Means (Lloyd, 1982). We apply each transformation on two settings; first, the semantic information channel is reduced, then concatenated with the sentence embedding vector. Alternatively, we apply the reduction on the concatenated, enriched vector itself.

4 Experiments

4.1 Datasets

We use the english version of the Multiling 2015 single-document summarization dataset (Giannakopoulos et al., 2015; Conroy et al., 2015) [1] for our experimental evaluation. The dataset is

[1] http://multiling.iit.demokritos.gr/pages/view/1516/multiling-2015

feature	train	test
document sentences	233	184.9
document summary sentences	77.9	13.5
document words	25.5	22.8
samples	6990	5546

Table 1: Details of the Multiling 2015 single-document summarization dataset. All values are averages accross documents, except from the number of samples.

built from wikipedia content, consisting of articles paired with a number of human-authored summaries. For each of 40 languages, 30 documents and summary sets are provided.

In this work, we focus on the English version of the dataset, due to our reliance on word embedding features, which are predomninantly available for the English language. In addition, we apply two preprocessing steps. First, we reformat the ground truth towards an extractive summarization setting, since the provided summaries are written from scratch. Specifically, we annotate source sentences with an extractive summarization binary label $l \in \{0, 1\}$ (e.g. 1 if it is a member of the extractive summary and 0 otherwise). This is accomplised via the following steps. First, for each provided summary sentence p_i, we rank source sentences $s \in S$ with respect to the n-gram overlap with p_i, after stopword filtering and excluding already positively-labelled sentences $s_j \in S : l_j = 1, i \neq j$. The top-ranked source sentence is matched to the ground truth summary sentence, and considered to be a member of the extractive summary. Secondly, in an effort to address the severe imbalance that results from the modifications of previous step (i.e. class 0 being 13 to 14 times more populous than class 1), we oversample positively labelled sentences for each document, arriving at a $2 : 1$ negative to positive ratio, at most.

After these steps, we end up with the final version of the dataset which is described in detail in table 1. Having a sentence-level label for summary meronymy, we can thus produce the final summary by concatenating the positively classified sentences. It should be noted that via this setting, evaluating candidate summaries with the dataset provided ground truth summaries implies a minimum performance penalty. This is reported in the results in the succeeding section 4.3.

4.2 Setup

We train embeddings with the word2vec CBOW variant using gensim (Rehurek and Sojka, 2011). We run the algorithm for 50 epochs, on a 10-word window, mantaining a minimum word frequency threshold of 2 occurences in the training text. We produce 50-dimensional embeddings via this process. In addition, we use the publicly available [2], 300-dimensional pre-trained fastext embeddings for the corresponding configuration.

To setup the deep neural classifier, we run a grid search on the number of layers (ranging from 1 to 5) and layer size (ranging from 64 to 2048) for a multilayer perceptron architecture, using a 5-fold validation scheme. This process illustrated a 5-layer architecture of 512-neuron layers as the best performing, and is the one we adopt for all subsequent experiments. This architecture is trained for 80 epochs, reducing the learning rate on an adaptive learning rate reduction policy and mantaining an early-stopping protocol of 25 epochs.

Using this learning framework, we test each candidate configuration using a 5-fold validated scheme, reporting mean measure values as the overall result. For all measures, the cross-fold variance stayed below $10e-4$ and is omitted. The Keras machine learning library [3] is used for building and training the neural models.

4.3 Results and discussion

Tables 2 and 3 present experimental results for the evaluation of semantic augmentation on word2vec and fastext embeddings, respectively. Each configuration is evaluated in terms of micro and macro F1 score (`mi-F` and `ma-F` columns, respectively), with respect to classification performance of the oversampled dataset (as detailed in 4.1). In addition, we measure Rouge-1 and Rouge-2 scores of the final composed summary (stiched together from positively classified input sentences) with respect to the hand-written ground truth summary provided in the dataset. Since the difference between the latter two guarantees a minimum error (see 4.1), we report the best possible performance in the `gt` configuration, depicting performance for each evaluation measure when sentence classification is perfect. In addition, via the `prob` configuration we report a probabilistic baseline classifier, which decides based on the label distribution

[2] https://fasttext.cc/
[3] https://keras.io/

in the training data. Moreover, token frequency-based baselines – namely bag-of-words (BOW) and TF-IDF (Salton and Buckley, 1988) – are reported in the `BOW` and `TF-IDF` rows. Lexical-only and semantically-augmented baseline runs are reported as `x` and `x-sem` respectively, where `x` ∈ [`w2v`, `fastext`]. Finally, the effect of each post-processing method on the semantically augmented baseline is illustrated, where a configuration of `+conf-N` denotes a vector post-processing method `conf` that produces `N`-dimensional vectors. The resulting vector dimension that is fed to the classfier is reported in the column `dim`, and the different semantic augmentation post-processing methods are denoted by `tc` – i.e. first transform the semantic channel, then concatenate to the embedding – and `ct` – i.e. concatenate the semantic vector to the lexical embedding, then apply the transformation.

Regarding word2vec trained embedings (table 2), we can see that introducing semantic information improves macro F1, Rouge-1, Rouge-2 performance. Compared with the bag-based baselines, we observe the word2vec CBOW embeddings yielding worse micro F1 performance than both bag approaches, but considerably better Rouge scores. In addition, the semantically enriched `w2v` configuration outperforms the bag approaches in macro-F1 score and the examined Rouge measures.

In general, we observe that micro-F1 scores appear to be less reliable measures in this setting, given the considerable large class imbalance of the dataset. This is apparent in the baseline `w2v` and `w2v-sem` baseline runs, however the effect is most pronounced in k-means configurations for dimensions greater than 50, where the best micro-F1 score is encountered, but the performance of all other metrics is degenerate. This is understandable, since cases where the classifier completely relies on the majority class (0, or "non-summary sentences" in our case), it can converge to a state characterized by a total lack of positively classified sentences. This in turn produces zero rouge scores and sub-chance macro-averaged F1 scores, which is the case observed for these configurations. The best-performing configuration turns out to be LSA with 500-dimensional vectors, with regard to Rouge-1 and Rouge-2 scores, with the 100-dimensional PCA configuration performing best in terms of macro F1.

Regarding comparison between the two post-processing strategies, we can observe that `tc` appears to be working slightly better when measuring micro-F1 scores, but in terms of macro-F1 and Rouge scores, concatenating prior to post-processing works considerably better. This is not surprising, as the transformation of the bimodal vector into a common, shared space can be expected to be a far more efficient fusion of the lexical and semantic channels, compared to simple concatenation.

Regarding fastext-based runs, a similar baseline performance is observed. Bag-based baselines achieve best micro-F1 score, but inferior results in all other measures. Similarly to word2vec, the lexical-only fastext run achieves better F1 scores, however the semantically enriched embedding fares far better in terms of Rouge-1 and Rouge-2 performance. Likewise, similar behavior is observed with regard to post-processing and concatenation order and the usefulness of the micro-F1 score compared to the other measures. Notably, the 50-dimensional LSA performs well with the `tc` strategy, while an analogous degenerate behaviour is evident with the K-means configurations. As in the word2vec run, the 500-dimensional LSA produces the best macro-F1 and Rouge scores.

Comparing the word2vec and fastext-based runs, we can observe the word2vec configurations (trained on the target dataset from scratch) achieve better Rouge-1 and Rouge-2 scores than the pre-trained fastext embeddings, on the best configurations of both baseline and best performing post-processed configurations (500-dimensional LSA).

In light of these results, we return to the research questions stated in the beginning of this document.

- **Can the introduction of semantic information in the network input improve extractive summarization performance?**

  It appears that the introduction of semantic information can introduce benefits to the extractive summarization pipeline. This is illustrated by the Rouge scores, which are considerably improved in the augmented configurations, for both embeddings examined. On the contrary, micro / macro-F1 results are either not significantly affected or can even deteriorate. However, as discussed above, we argue that the severe class imbalance of the dataset

config	dim	mi-F		ma-F		Rouge-1		Rouge-2	
gt	N/A	1.000		1.000		0.414		0.132	
prob	N/A	0.871		0.501		0.051		0.009	
BOW	15852	0.9254		0.5131		0.094		0.017	
TF-IDF	15852	0.9260		0.5122		0.085		0.015	
w2v	50	0.923		0.508		0.151		0.027	
w2v-sem	10292	0.906		0.519		0.260		0.048	
config	dim	tc	ct	tc	ct	tc	ct	tc	ct
+lsa-50	100	0.9225	0.9214	0.5223	0.5222	0.166	0.195	0.030	0.036
+lsa-100	150	0.9202	0.9207	0.5164	0.5217	0.188	0.202	0.038	0.038
+lsa-250	300	0.9197	0.9165	0.5198	0.5289	0.181	0.246	0.037	0.040
+lsa-500	550	0.9218	0.9053	0.5190	0.5337	0.159	**0.305**	0.030	**0.059**
+pca-50	100	0.9208	0.9101	0.5195	0.5329	0.193	0.242	0.039	0.049
+pca-100	150	0.9207	0.9141	0.5206	**0.5349**	0.178	0.234	0.036	0.047
+pca-250	300	0.9217	0.9146	0.5217	0.5250	0.171	0.237	0.035	0.044
+pca-500	550	0.9223	0.9107	0.5202	0.5254	0.161	0.255	0.032	0.049
+kmeans-50	100	0.9089	0.9257	0.5267	0.4821	0.252	0.018	0.056	0.005
+kmeans-100	150	0.9028	**0.9272**	0.5107	0.4811	0.133	0.000	0.028	0.000
+kmeans-250	300	**0.9272**	**0.9272**	0.4811	0.4811	0.000	0.000	0.000	0.000
+kmeans-500	550	**0.9272**	**0.9272**	0.4811	0.4811	0.000	0.000	0.000	0.000

Table 2: Experimental results on the MultiLing 2015 MSS dataset using 50-dimensional word2vec embeddings. Bold values indicate maxima across rows for that column. Underlined values correspond an improvement over the counterpart configuration (`tc` versus `ct`, or `x` versus `x-sem`).

config	dim	mi-F		ma-F		Rouge-1		Rouge-2	
gt	N/A	1.000		1.000		0.414		0.132	
prob	N/A	0.871		0.501		0.051		0.009	
BOW	15852	0.9254		0.5131		0.094		0.017	
TF-IDF	15852	0.9260		0.5122		0.085		0.015	
fastext	300	0.923		0.517		0.156		0.029	
fastext-sem	10542	0.919		0.516		0.204		0.043	
config	dim	tc	ct	tc	ct	tc	ct	tc	ct
+lsa-50	350	0.9167	0.9214	0.5231	0.5195	0.206	0.182	0.038	0.032
+lsa-100	400	0.9200	0.9212	0.5196	0.5224	0.171	0.189	0.032	0.036
+lsa-250	550	0.9237	0.9134	0.5221	0.5370	0.145	0.278	0.031	0.053
+lsa-500	800	0.9243	0.9083	0.5201	**0.5373**	0.128	**0.296**	0.025	**0.056**
+pca-50	350	0.9186	0.9145	0.5205	0.5319	0.182	0.234	0.036	0.045
+pca-100	400	0.9208	0.9160	0.5187	0.5369	0.160	0.230	0.037	0.044
+pca-250	550	0.9233	0.9146	0.5210	0.5286	0.189	0.229	0.038	0.045
+pca-500	800	0.9239	0.9096	0.5223	0.5261	0.152	0.255	0.032	0.047
+kmeans-50	350	0.8995	0.9238	0.4928	0.4833	0.071	0.022	0.018	0.006
+kmeans-100	400	0.8903	**0.9272**	0.4897	0.4811	0.071	0.000	0.018	0.000
+kmeans-250	550	**0.9272**	**0.9272**	0.4811	0.4811	0.000	0.000	0.000	0.000
+kmeans-500	800	**0.9272**	**0.9272**	0.4811	0.4811	0.000	0.000	0.000	0.000

Table 3: Experimental results on the MultiLing 2015 MSS dataset using 300-dimensional fasttext embeddings. Bold values indicate maxima across rows for that column. Underlined values correspond an improvement over the counterpart configuration (`tc` versus `ct`, or `x` versus `x-sem`).

makes these measures less indicative of system performance, compared to Rouge.

- **Does the semantic augmentation process benefit via dimensionality reduction post-processing methods?**

  The augmentation process can improve with post-processing methods. This is expected, since the sparse bag-based semantic vectors are bound to contain noise and/or redundant and overlapping information that will affect the learning model further down the summarization pipeline. For both embeddings examined, such configurations improve upon the baseline and achieve the best scores, for all evaluation measures included.

  LSA-based transformations achieve top Rouge performance for both embeddings covered, as well as top F1 scores for the fastext embedding, with its frequency-based decomposition appearing to work better than PCA analysis. On the contrary, K-means clustering mostly failed to capture underlying semantic content into meaningful groups, especialy for higher dimensions examined. Additionally, the post-processing transformation methods work best mostly when applied to the concatenated lexical and semantic vectors, rather than transforming the semantic information alone and then conatenating.

  Apart from the specific research questions, it is notable that the large class imbalance has to be carefully handled, as – even with the dataset oversampling measures taken – the sentence classifier can converge into degenerate cases, as was the case with the higher dimensional configurations of K-means.

  At this point, we note that since our system does not account for selected sentence order, we limit our comparison of each approach to only the `gt` configuration, rather than the human-authored summaries; even for cases with perfect classification performance, the results are far from optimal (e.g. Rouge 1, Rouge 2 scores of 1.0) since there is no guarantee that sentence order is preserved in the extractive ground truth generation, detailed in 4.1. This introduces an upper bound to performance and prevents meaningful comparison to related work. Instead, this study illustrates the contribution of semantic information to the pipeline, as illustrated above.

  As a last note, we compare our results with respect to the unaltered, human-written summaries – i.e. which are not composed of input sentences as per the extractive setting, after reiterate that our preliminary approach does not take into account sentence order or target length. First, the `gt` extractive ground truth we generated achieves an Rouge-1 and Rouge-2 score of 0.245 and 0.57 respectively, effectively serving as an upper bound for our performance. The best-performing 500-dimensional LSA configuration for word2vec trained embeddings performs at 0.196 and 0.015 for Rouge-1 and Rouge-2, respectively, and 0.191, 0.014 for fasttext. These results fall short of the system performance levels on previous MultiLing community tasks (Conroy et al., 2015), however the goal of this investigation was solely to illustrate the utility of the semantic component; future work (outlined below) plans on addressing this issue and align our results toward related work comparability.

5 Conclusions

In this work, we investigated the contribution of semantically enriching word embedding-based approaches to extractive summarization. Pre-trained embeddings as well as embeddings trained from scratch on the target dataset were utilized. For the semantic channel, frequency-based concept information from Wordnet is extracted, post-processed with a range of feature transformation and clustering methods prior or after concatenation with the lexical embeddings. A wide evaluation was performed on multiple configuration combinations and transformation dimensions, using micro/macro F1 and Rouge-1/Rouge-2 scores. Initial results show semantic such augmentation approaches can introduce considerable benefits to baseline approaches in terms of macro F1, Rouge-1 and Rouge-2 scores, with micro-F1 deemed inadequate for highly imbalanced problems such as the extractive summarization setting examined here. LSA-based decomposition works best out of the variants examined, outperforming PCA and K-means post-processing in terms of Rouge. In the future, more sophisticated transformation methods could be explored, such as encoder-decoder

schemes via recurrent neural networks (Hochreiter and Schmidhuber, 1997), dynamically fusing word embeddings into a sentence encoding and eliminating the need for word averaging in sentence-level vector generation. Alternatively, sequence-based classification could be explored in a similar fashion. Moreover, higher transformation dimensions could be covered, given the best configuration examined lied on the highest end of the exmained range (500) and additional semantic resources can be utilized, via the bag-based approach used in this study, or by alternative methods of semantic vector generation (Faruqui et al., 2014). Finally, the natural next step in our work would be the application of our semantic augmentation approach with a sentence ranking and a target length constraint mechanisms, in order to make the results of pipeline fairly comparable to related summarization systems.

References

Abdelkrime Aries, Djamel Eddine Zegour, and Khaled Walid Hidouci. 2015. Allsummarizer system at multiling 2015: Multilingual single and multi-document summarization. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 237–244.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 86–90.

Kedar Bellare, Anish Das Sarma, Atish Das Sarma, Navneet Loiwal, Vaibhav Mehta, Ganesh Ramakrishnan, and Pushpak Bhattacharyya. 2004. Generic text summarization using wordnet. In *LREC*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, pages 1247–1250.

Peter F Brown, Peter V DeSouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Comput. Linquistics* 18(1950):467–479. http://www.aclweb.org/anthology/J92-4003.

John M. Conroy, Jeff Kubina, Peter A. Rankel, and Julia S. Yang. 2015. *Multilingual Summarization and Evaluation Using Wikipedia Featured Articles*, chapter Chapter 9, pages 281–336.

CHENGHUA Dang and XINJUN Luo. 2008. Wordnet-based dcument summarization. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. World Scientific and Engineering Academy and Society, 7.

Scott Deerwester, Susan T ST Dumais, George W GW Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* 41(6):391. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6¡391::AID-ASI1¿3.0.CO;2-9.

Zakaria Elberrichi, Abdelattif Rahmoun, and Mohamed Amine Bentaalah. 2008. Using wordnet for text categorization. *International Arab Journal of Information Technology (IAJIT)* 5(1).

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv Prepr. arXiv1411.4166* (i). https://doi.org/10.3115/v1/N15-1184.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 961–968.

George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. 2015. Multiling 2015: multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 270–274.

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence* 2(3):258–268.

Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. 2005. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer* 27(2):83–85.

Sepp Hochreiter and Jurgen Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8):1–32. https://doi.org/10.1162/neco.1997.9.8.1735.

Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

Chihli Hung and Stefan Wermter. 2004. Neural network based document clustering using wordnet ontologies. *International Journal of Hybrid Intelligent Systems* 1(3-4):127–142.

Anjali Ganesh Jivani, Others, and Ganesh Jivani Anjali. 2011. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl* 2(6):1930–1938. https://www.researchgate.net/profile/Anjali_Jivani/publication/284038938_A_Comparative_Study_of_Stemming_Algorithms/links

Ian Jolliffe. 2011. Principal component analysis. In *International encyclopedia of statistical science*, Springer, pages 1094–1096.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification https://arxiv.org/pdf/1607.01759.pdf.

S Katz. 2003. Estimation of probabilities from sparse data for the language model component of a speech recognizer (February):2–4.

Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. 2001. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 349–357.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. pages 1188–1196.

Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Lecun Y., Bengio Y., and Hinton G. 2015. Deep learning. *Nature* 521(7553):436–444. https://doi.org/10.1038/nature14539.

Lei Li, Yazhao Zhang, Liyuan Mao, Junqi Chi, Moye Chen, and Zuying Huang. 2017. Cist@ clscisumm-17: Multiple features based citation linkage, classification and summarization. In *BIRNDL@ SIGIR (2)*. pages 43–54.

Marina Litvak and Mark Last. 2013. Multilingual single-document summarization with muse. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*. pages 77–81.

Ying Liu, Peter Scheuermann, Xingsen Li, and Xingquan Zhu. 2007. Using wordnet to disambiguate word senses for text classification. In *international conference on computational science*. Springer, pages 781–789.

Elena Lloret and Manuel Palomar. 2009. A gradual combination of features for building automatic summarisation systems. In *International Conference on Text, Speech and Dialogue*. Springer, pages 16–23.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE Trans. Inf. theory* 28(2):129–137.

Trevor N Mansuy and Robert J Hilderman. 2006. Evaluating wordnet features in text classification models. In *FLAIRS Conference*. pages 568–573.

Victoria McCargar. 2004. Statistical approaches to automatic text summarization. *Bulletin of the American Society for Information Science and Technology* 30(4):21–25.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. pages 404–411.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM* 38(11):39–41.

Tatsunori Mori. 2002. Information gain ratio as term weight: the case of summarization of ir results. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. *Proc. Tenth Int. Work. Artif. Intell. Stat.* pages 246–252. https://doi.org/10.1109/JCDL.2003.1204852.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005* 101.

Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Brazilian Symposium on Artificial Intelligence*. Springer, pages 205–215.

Alok Ranjan Pal and Diganta Saha. 2014. An approach to automatic text summarization using wordnet. In *2014 IEEE International Advance Computing Conference (IACC)*. IEEE, pages 1169–1173.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Matt Post and Shane Bergsma. 2013. Explicit and implicit syntactic features for text classification. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 866–872.

D Radev, H Jing, and M Budzikowska. 2000. Centroid-based summarization of multiple documents: Clustering, sentence extraction, and evaluation. In *Proceedings of the ANLP/NAACL-2000 Workshop on Summarization*.

Radim Rehurek and Petr Sojka. 2011. Gensim—statistical semantics in python. *statistical semantics; gensim; Python; LDA; SVD* .

M Rodriguez, J Hidalgo, and B Agudo. 2000. Using wordnet to complement training information in text categorization. In *Proceedings of 2nd International Conference on Recent Advances in Natural Language Processing II: Selected Papers from RANLP*. volume 97, pages 353–364.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* 24(5):513–523.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18(11):613–620.

Sam Scott and Stan Matwin. 1998. Text classification using wordnet hypernyms. *Usage of WordNet in Natural Language Processing Systems* .

Sam Scott and Stan Matwin. 1999. Feature engineering for text classification. *ICML* 99:379–388. https://pdfs.semanticscholar.org/6e51/8946c59c8c5d005054af319783b3eba128a9.pdf.

Marta Vicente, Oscar Alcón, and Elena Lloret. 2015. The university of alicante at multiling 2015: approach, results and further insights. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 250–259.

Yiming Yang. 1997. A comparative study on feature selection in text categorization http://www.surdeanu.info/mihai/teaching/ista555-spring15/readings/yang97comparative.pdf.

Jaya Kumar Yogan, Ong Sing Goh, Basiron Halizah, Hea Choon Ngo, and C Puspalata. 2016. A review on automatic text summarization approaches. *Journal of Computer Science* 12(4):178–190.