# Sentence Embeddings as an intermediate target in end-to-end summarisation

**Maciej Zembrzuski**
trivago N.V. / Düsseldorf, Germany
maciej.zembrzuski@trivago.com

**Saad Mahamood**
trivago N.V. / Düsseldorf, Germany
saad.mahamood@trivago.com

## Abstract

Current neural network-based methods to the problem of document summarisation struggle when applied to datasets containing large inputs. In this paper we propose a new approach to the challenge of content-selection when dealing with end-to-end summarisation of user reviews of accommodations. We show that by combining an extractive approach with externally pre-trained sentence level embeddings in an addition to an abstractive summarisation model we can outperform existing methods when this is applied to the task of summarising a large input dataset. We also prove that predicting sentence level embedding of a summary increases the quality of an end-to-end system for loosely aligned source to target corpora, than compared to commonly predicting probability distributions of sentence selection.

## 1 Introduction

Document summarisation is the task in which texts are ingested and a shorter textual summary is produced. This task continues to receive considerable attention within the Natural Language Processing community for various information access purposes and comprises of two main paradigm forms: *abstractive* and *extractive*. Extractive summarisation focuses on the task of choosing salient or relevant sentences in a given corpora. Whilst abstractive summarisation involves re-writing a given corpora into a more concise form that summarises the input.

Most recent techniques for summarisation either *abstractive* or *extractive* have come to rely on neural network techniques with a particular focus on attempting to improve the quality of the generated summaries from long documents. This has included attempts to improve how and what text is summarised. For example, by trying to optimise extractive summarisation for the aspect of coherence (Wu and Hu, 2018), for evaluation metrics (Paulus et al., 2017; Kryściński et al., 2018; Chen and Bansal, 2018), the importance of a given sentence (Zhou et al., 2018), or by the application of a pre-trained Transformer (Vaswani et al., 2017) model as show by BERTSUM (Liu, 2019). For abstractive summarisation improvements in content-selection have come about by decoupling the task of selecting of salient content and then performing abstractive summarisation as a separate discrete step (Gehrmann et al., 2018).

However, most past work on summarisation has tended to focus on summarising newswire corpora. Work that has utilised user review corpora has either focused on generating review titles (Yang, 2016) or for describing the benefits or disadvantages of products (Kunneman et al., 2018).

In this paper we present the USEsum model for generating short unique selling point summaries from a corpora of user hotel reviews. We use a two-stage system that combines both extractive and abstractive summarisation techniques to enable salient content selection and to allow the generation of a summary over a large input corpora. In particular, this paper makes the following key contributions:

- We propose a new solution to the task of summarising user reviews.
- Demonstrate the value of representing semantic information by using pre-trained sentence embeddings through the use of the Universal Sentence Encoder (Cer et al., 2018).
- Show that the use of angle measurement between sentence embeddings is a good metric for comparing the semantics of generated summary text.

## 2 Related Work

Past approaches to the problem of summarising multiple-documents have tend to be extractive in

nature with the most important sentences extracted and then optionally compressed to form a summary. Dorr et al. (2003), for example when generating newspaper headlines used a non-neural approach. By extracting nouns and verb phrases from the first sentence and of a news article and then using an iterative algorithm to compress the sentence to the length of a headline. Neural approaches such as the one used by Durrett et al. (2016) have take the same idea, but also learn a model to select sentences and compress them.

More recent approaches in document summarisation have focused on the challenge generating high quality summaries using artificial neural networks. One method to this problem has been the application of deep reinforcement learning to take into account the aspect of coherence in the generated generated summaries (Wu and Hu, 2018) or to directly optimise for evaluation metrics (Paulus et al., 2017; Kryściński et al., 2018; Chen and Bansal, 2018) to improve the quality of the generated summarised output.

Alternative approaches for improving the quality of extractive summarisation have included the use of a pre-trained BERT (Devlin et al., 2018) model such as the BERTSUM system (Liu, 2019). This approach predicts a score for each sentence directly in addition to using the Transformer model on-top of sentence embeddings. This has resulted in improved ROUGE scores when tested on *CNN/Daily Mail* corpora in comparison to other state-of-the-art extractive systems. Systems such as NEUSUM (Zhou et al., 2018) demonstrate a successful approach in the use of model sentence importance scores and thus improving the quality of its extractive summarisation. Additionally, both the REFRESH (Narayan et al., 2018) and DCA (Celikyilmaz et al., 2018) systems also show that the summarisation task can improve with the application of reinforcement learning.

Abstractive summarisation attempts to to produce a summary, which may contain aspects that were not part of the original input. The use of a neural language model in combination with a beam search decoder can result in the generation of more accurate summaries (Rush et al., 2015). Nevertheless, one of the problems with end-to-end neural network-based methods for abstractive summarisation tend to perform poorly at content selection (Gehrmann et al., 2018). This is because such models can include content that is neither relevant or salient in the generated summary. One technique to fix this deficiency has been the development of a two-stage generation approach. Firstly by performing content-selection through the use of a bottom-up selector that selects salient phrases in the source document and secondly by performing the step of generating of abstractive summaries using a standard neural model with the given selection mask. This two step-approach has show to outperform other alternative approaches that have attempted to perform content-selection as part of an end-to-end model (Gehrmann et al., 2018).

An alternative to the two-stage abstractive summarisation technique proposed by Gehrmann et al. (2018), is the use of a discourse parser to obtain a discourse tree of user products reviews. Selection of content is driven by the use of the PageRank algorithm to select a sub-graph of the most important aspects, which in combination with a template based Natural Language Generation is used to generate an abstractive summary (Gerani et al., 2014).

An additional challenge is the ability to generate high quality abstractive summaries when faced with large documents or summaries without including repetitive or incoherent phrases. Work by Paulus et al. (2017) has shown that with the use of a intra-attention model that pays attention to both previously used input tokens in the encoder and the words already generated by the decoder it is possible to generate higher quality and more readable abstractive summaries (2017).

Most past work related to utilising user reviews for summarisation has focused on predicting review titles (Yang, 2016). The work of Kunneman et al. (2018), for example, focused on extracting the positives and negatives of products from user reviews. This approach required the reviews to be formatted in a specific structure, which is not present in the analysed dataset. Overall, the reviews summarisation task differs significantly from news summarisation due to the phenomena of opinion shifting (Pecar, 2018) and the lack of document structure.

## 3 Unique Selling Point dataset

The USEG dataset[1] represents the problem of describing the unique characteristics of a hotel based on user reviews. Each target hotel description consists of one sentence and the related reviews contain up to 800 sentences per hotel.

[1] USEG — https://github.com/useg-data/useg-data

This problem can be seen as a multi-document summarisation challenge as each user reviews is independent of each other. On the other hand, reviews are short and written in various styles. This limits the possibility of benefiting from document structure. In this approach the reviews were merged into a single document per hotel. Therefore, the summarisation tasks depends more on extracting the most interesting information, irrespective of where the information is located in the input text. We see this task as a special kind of single document summarisation challange where there is no document structure.

The task is also unique in the way that the summaries are not necessarily meant to cover the most commonly represented features or amenities of a hotel. For example, a hotel can be well located, but what makes it unique is the fact that it contains a rooftop swimming pool. The "compression rate" is relatively high compared to commonly used datasets (e.g. *CNN/Daily Mail*, *DUC 03/04*, *Gigaword*, etc.). This dataset requires the analyse of up hundreds of sentences to generate a single sentence. The style of descriptions also differ significantly from the reviews' style, therefore the word overlap between the descriptions and the user reviews is smaller than in commonly used datasets. In the USEG dataset only a third of the summaries overlap sufficiently with source reviews that the NEUSUM (2018) approach could use them for training, based on the ROUGE score (Lew, 2004). However, when similarities were calculated using pre-trained sentence embeddings (Cer et al., 2018), two thirds of the descriptions contained information covered by reviews that were formulated using similar but different expressions. The remaining third of summaries don't contain information which is reflected in aligned user reviews. Nevertheless, this is still valid for training the extractive approach in proposed architecture.

## 4 Proposed Approach

Due to the nature of the USEG dataset, where the target sentences differ significantly in style and length from input sentences, the task requires an abstractive approach to generate the USP summaries. On the other hand, due to large amount of input sentences, it is easier to retrieve the requisite information about the unique characteristics of a given accommodation in an extractive manner from the user reviews. Therefore, the USEsum system consists of both approaches. An extractive model firstly selects the top three sentences from user reviews while a secondary abstractive approach predicts the final summary description. The inspiration for selecting three sentences for further processing is the fact that selection of the three initial sentences (LEAD 3 (2016)) is sufficient to constitute a strong baseline in *CNN/Daily Mail* summarisation.

To further limit processing on the large amount of input sentences by the extractive model, they are pre-processed in such a way that each input sentence is represented by a semantically meaningful vector, furtherly referred to as a sentence embedding. The sentence to vector calculation is done using the Universal Sentence Encoder (Cer et al., 2018). The extractive model predicts sentence embedding vectors in the same space as the input vectors. This allows for the selection of the most relevant sentences from the given input, which are concentrated around a common concept. After predicting the sentence embedding, it is compared with the embeddings of the input sentences and the three most similar input sentences are selected. This proposed system was implemented using a custom adaptation of OpenNMT (Klein et al., 2017). Figure 1 illustrates an abstract overview of how the USEsum system performs end-to-end summarisation.

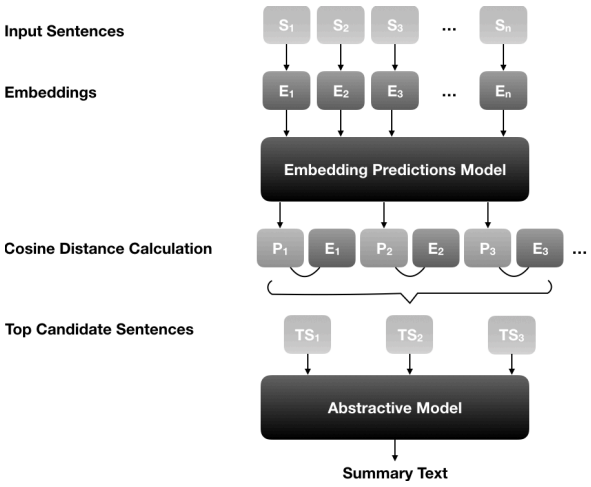


Figure 1: USEsum system diagram for generating USP abstractive summaries

The source code and models used for the USP summary generation task are available online[2].

[2]USP summary generation implementation code and models — https://github.com/USE-sum/usesum

### 4.1 Problem definition

Extractive document summarisation aims to select the most salient sentences from a given document. Each document $D_j$ in set of documents to summarise, contains $L$ sentences $D_j = (S_1, S_2, ..S_i.., S_L)$. The extractive summarisation system tries to identify the subset of $D_j$ which in the USEsum setting will be further processed by the abstractive summarisation system.

During the training phase of the extractive system, the sentence embedding of the reference summary $y$ and the embedding $x_i$ of the $i-th$ processed sentence are given. Both $y$ and $x_i$ are semantically meaningful sentence embeddings, calculated by Universal Sentence Encoder (Cer et al., 2018). The goal of training is to learn a scoring function which will predict the influence of the currently processed input embedding $x_i$ on the final embedding prediction $\hat{y}_L$ of the document $D_j$.

The proposed approach follows Maximal Marginal Relevance (MMR) method proposed by Carbonell and Goldstein (1998). The MMR method tries to maximise the relative gain given previously extracted information. In the proposed approach the information gain at $i-th$ step is measured by comparing angles between sentence embeddings $y$ and $\hat{y}_i$.

The USEsum model is trained to maximize a scoring function $g(\cdot)$ of the information gain represented by:

$$g(x_i|\hat{y}_{i-1}) = angle(y, \hat{y}_i) - angle(y, \hat{y}_{i-1}) \quad (1)$$

At each processing step $i$ the summarisation system estimates the influence of the processed input sentence embedding $x_i$ on the predicted document embedding $\hat{y}_L$.

### 4.2 Extractive approach

Considering the lack of structure of the input documents in the dataset, we have employed the use of the Universal Sentence Encoder (Cer et al., 2018) to transform the documents represented as a sequence of sentences into a sequence of sentence embeddings. These sentence embeddings are further encoded by the bi-directional LSTM (Hochreiter and Schmidhuber, 1997) resulting in encoder memory states at $i-th$ step represented by $s_i$. Subsequent processing by the decoder is defined by the following equations:

$$fg = \sigma(W_{fg}([s_i, \alpha, \beta]) \quad (2)$$

$$ig = \sigma(W_{ig}([s_i, \alpha, \beta]) \quad (3)$$

$$og = \sigma(W_{oxg}(x_i) + W_{ogy}(\hat{y}_{i-1})) \quad (4)$$

$$ct = fg * \hat{y}_{i-1} + ig * tanh(W_{ct}(x_i)) \quad (5)$$

$$ht = og * \sigma(ct) \quad (6)$$

$$score = \sigma(W_{sc}(ht)) \quad (7)$$

$$\hat{y}_i = \hat{y}_{i-1} - score * (\hat{y}_{i-1} - x_i) \quad (8)$$

Where $W$ are learnable weights, $\sigma$ is a sigmoid transformation, $\hat{y}_i$ is the prediction at $i$-th processing step, and $x_i$ is the sentence embedding of the $i-th$ input sentence. $\alpha$ is the cosine similarity between $\hat{(y)}_{i-1}$ and $x_i$ . $\beta$ is the cosine similarity between $\hat{(y)}_{i-1}$ and vector $\omega_j$ representing all sentences in a document. $\omega_j$ is calculated by the following equation:

$$\omega = \sum_{i=0}^{N_j} x_i \quad (9)$$

For processing of the first sentence, $\hat{y}_{i-1}$ is initialised as:

$$\hat{y}_{-1} = tanh(W(tanh(\omega_j))) \quad (10)$$

Scoring the influence of the difference between $x_i$ and $\hat{y}_{i-1}$ aims at mitigating the phenomena of information redundancy and dealing with the fact that the most popular information doesn't have to be related the target. The decision to focus on the use of a RNN instead of using a transformer was motivated by the very long inputs, which resulted in reaching GPU memory constraints.

#### 4.2.1 Objective function

A training step is defined as the processing of a single input sentence and predicting new estimation of the target vector. The loss calculation at each training step depends on the arcus cosine value of cosine similarity between predicted embedding vector and the target embedding. The final loss at each step is the difference between the current and previous step distances from the target. Therefore, there is no loss during the first training step. The loss is negative when an improvement in comparison to previous estimation has occurred. We found that training the model with both positive and negative loses improves performance.

The model is expected to optimize the information extraction for each sentence. Predicting the final vector is a side effect of this approach.

$$loss_i = acos(\hat{y}_{i-1}, target) - acos(\hat{y}_i, target) \quad (11)$$

Where $\hat{y}_i$ is the sentence embedding prediction at step $i$. $acos$ stands for arcus cosine similarity. This approach allows for utilisation of training examples where there is a week alignment between sources and target. The model can still learn which sentences are comparatively more informative, even if none of them matches the target perfectly.

### 4.3 Abstractive summarisation

The predicted embedding $\hat{y}$ is further used to extract three sentences which are an input for abstractive summarisation. These sentences are selected based upon the angle similarity between each of the input sentence embeddings $x_i$ and the predicted summary embedding $\hat{y}_L$. For abstractive summarisation we have employed the use of a Transformer (2017) in a standard configuration with additional word features such as word lemma, POS tags, NER tags, and dependency type. This was inspired by the work of Nallapati et al. (2016). Word embeddings are initialised using 300 dimensions of GloVe[3] (Pennington et al., 2014) and the copy attention proposed by Gu et al. (2016). The loss function was altered by the usage of focal loss (Lin et al., 2017) with the default parameter $\gamma$=*2*. We also updated beam search in the inference phase to recognise potential named entities in the generated summary candidates, to penalise entities which were not present in source text and promote weights of the candidate entities present in the input. As discussed more broadly in the Results section, this approach helped to solve problematic cases such as a wrong city would be mentioned in a generated summary. The penalty and promotion factors were estimated manually on validation set. The beam scores of candidate words recognised as named entities, not present in the input, are multiplied by a factor of *50*, and the ones present by a factor of *0.4*. The abstractive model is trained independently from the extractive model. During training phase, the model is provided with three of the most similar sentences measured by the angle similarity between these sentences' embeddings and target summary embedding.

## 5 Evaluation

To evaluate the proposed solution for generating USP summaries we have chosen the USEG dataset. This dataset is to the best of our knowledge the only publicly available dataset for summarising user reviews at a product level. This dataset differs considerably from usually used datasets for document summarisation, such as *CNN/Daily Mail*, *NYTimes*, *DUC-03*, and *DUC-04* (Hermann et al., 2015; Sandhaus, 2008; Over and Yen, 2003, 2004). The main features differentiating the USEG dataset from the commonly recognised datasets are the following aspects:

- Lack of document structure.
- Random distribution of information.
- Single sentence summaries.
- Low word overlap between source and target texts.

We have optimised the model to tackle these inherit challenges in the USEG dataset due to consisting of user generated reviews of accommodations. Adapting this approach for other datasets would required significant changes in architecture, which was considered out of scope of for this project due to time constraints. The changes would consist of adaptations to allow the model to benefit from the presence of a document structure and to be able to generate coherent multi-sentence summaries.

In addition to choice of dataset, we have have also chosen commonly recognised metrics for evaluating NLP based summarisation implementations such as ROUGE-L, BLEU, and METOR. We also used cosine similarity between sentence embeddings of generated summaries and targets as this would measure the semantic similarity between source and target irrespective of the words chosen. This is because in the proposed solution we have utilised the sentence embeddings calculated by the Universal Sentence Encoder (Cer et al., 2018) and these embeddings are semantically meaningful to make the cosine similarity a good measure for comparing similarity.

To evaluate USEsum model, we compare it with pre-existing competitive summarisation systems that have utilised the *CNN/Daily Mail* dataset. These systems were chosen on the basis of their performance on *CNN/Daily Mail* dataset, whether we were able to find their implementation code, and whether we could adapt the systems within the given time constraints to the USEG dataset. The chosen extractive systems include REFRESH

[3] GloVe — https://nlp.stanford.edu/projects/glove/

(Narayan et al., 2018), BERTSUM (Liu, 2019), and NEUSUM (Zhou et al., 2018). For these extractive models, we made minor adaptations to enable processing of USEG dataset. Maximal input sentence size for NEUSUM was *800* sentences. For BERTSUM we retained a *512* input word limit. This limited its performance due to the fact it could only process a fraction of the input data. However, changing this parameter would require deeper changes in it's implementation code. We added a baseline Universal Sentence Encoder (Cer et al., 2018) approach (further: BASELINE) where all the input sentence embedding vectors of a document were summed to $\omega$ vector as defined in equation 9. Furthermore, the sentence with the embedding that is most similar to the $\omega$ is selected.

For performing end-to-end comparisons the outputs of all the extractive models were processed by a common abstractive model.

For a comparison with an alternative end-to-end abstractive summarisation model we also trained a RNN model using OpenNMT. This was done using the pre-existing BOTTOM-UP abstractive summarisation approach (Gehrmann et al., 2018) from the documentation provided on the OpenNMT website[4]. The embeddings were initialized with *300* dimensional GloVe pre-trained embeddings. The encoder hidden size was *512* and decoder *1024*. To make the comparison more fair, the input words are augmented with the same features as described in section 4.3.

## 6 Results

To evaluate the effectiveness of USEsum, we performed several independent experiments, aiming to evaluate the following aspects:

- Evaluating the quality of the extractive summarisation.
- Evaluating the quality of the summary generated end-to-end.
- Estimating the effect of beam search input word promotions as described in section 4.3.

The inferred results for all of the experiments are available online[5].

[4]OpenNMT Summarisation Documentation — http://opennmt.net/OpenNMT-py/Summarization.html

[5]Experimental Results — https://github.com/USEsum/usesum/tree/master/results

### 6.1 Extractive summarisation experiment

The first experiment was to compare the quality of prediction of the best sentence selected by extractive models. Table 1 presents the results of the first experiment. The models selected for comparison are described in section 5. This experiment shows interesting phenomena; NEUSUM outperforms other approaches in all metrics except for cosine similarity between embeddings. The latter is interesting as it is the only metric which measures the semantic similarity of sentences, irrespective of the words used. As the final experiment in section 6.2 shows the end-to-end USEsum model outperforms other end-to-end approaches, which rely on extractive models for all metrics. This suggests the high importance of cosine metrics in this case.

### 6.2 End to end summarisation experiment

For the end-to-end approach, aiming at generating the desired summaries, we compare different extractive based models by combining them with the same abstractive model to obtain the final summary. To compare, we also include the BOTTOM-UP (Gehrmann et al., 2018) abstractive end-to-end model. All the models generated ten summary candidates. To select the best candidate, we used a simple heuristics of comparing each candidate sentence embedding with the embedding of the three input sentences for the abstractive model. For the BOTTOM-UP model the whole document was used as the input, therefore the candidates were compared with the embedding of the whole document. For additional comparison we also used a BOTTOM-UP model in which only the first sentence candidate is chosen as the final summary. This system is referred to as BOTTOM-UP $1^{st}$. The results of the end-to-end experiment are shown in table 2.

Both BERTSUM and REFRESH omit some test cases where they were unable to decide which sentence to choose. In total, the number of omitted cases amounted to two for BERTSUM and six for REFRESH. The comparison between BOTTOM-UP and BOTTOM-UP $1^{st}$ shows that using the Universal Sentence Encoder for selecting the best candidate, instead of picking the top output from the system, is beneficial. This post selection of candidates also improved results for all other models in additional experiments, which is not listed here for brevity. The USEsum model seems to outperform other models considering all metrics

| | BLEU | ROUGE-L | METEOR | Cosine Similarity |
|---|---|---|---|---|
| BASELINE | 0.0027 | 0.0632 | 0.0358 | 0.3874 |
| BERTSUM $1^{st}$ sentence | 0.0011 | 0.0503 | 0.0275 | 0.3502 |
| NEUSUM $1^{st}$ sentence | **0.0055** | **0.0846** | **0.0539** | 0.4149 |
| REFRESH $1^{st}$ sentence | 0.0004 | 0.0459 | 0.0227 | 0.2965 |
| USEsum $1^{st}$ sentence | 0.003 | 0.0761 | 0.0474 | **0.4493** |

Table 1: Extractive summarisation results

| | BLEU | ROUGE-L | METEOR | Cosine Similarity |
|---|---|---|---|---|
| BASELINE | 0.0063 | 0.1030 | 0.0448 | 0.489 |
| BERTSUM | 0.0040 | 0.1071 | 0.0414 | 0.4924 |
| BOTTOM-UP | 0.0188 | 0.1427 | 0.0543 | **0.5132** |
| BOTTOM-UP $1^{st}$ | 0.0153 | 0.1213 | 0.0468 | 0.4937 |
| NEOSUM | 0.0208 | 0.1217 | 0.0535 | 0.4866 |
| REFRESH | 0.0044 | 0.0948 | 0.0379 | 0.4559 |
| USEsum | **0.0225** | **0.1479** | **0.0602** | 0,5115 |

Table 2: End to end summarisation results

with the exception of cosine similarity where the BOTTOM-UP model performs best. The human analysis of the results gives more insights into the results and cosine metrics.

### 6.3 Human evaluation of the end-to-end experiment

In addition to the evaluation with automatic metrics, a manual evaluation was also conducted to estimate the quality and semantics of outputs generated by the BOTTOM-UP, NEUSUM, and USEsum models in comparison to the target summary. Table 3 illustrates the example outputs from these three systems.

The predictions from each system were given to human evaluators who performed an intrinsic evaluation for a hundred outputs on each system. The evaluators marked the number of outputs that were grammatically correct and whether the output covered the semantics of the target or not. Unlike the binary ratings assigned for grammatical correctness, if the output text partially covered the target semantics, for example mentioning one amenity or facility in a given target but not others, then the output text could be awarded half a point.

Table 4 illustrates the results of this evaluation between the three systems. Whilst, the BOTTOM-UP system performed best in terms of grammatical correctness, USEsum was better than other systems for semantic similarity with the intended target text. The BOTTOM-UP system learned to use several popular phrases and generated outputs by combining these phrases. For example, it mentioned the phrase "pet friendliness" in 96 predictions whereas there were only 9 in the target summaries. This approach helps in keeping grammatical correctness, however it also results in predicting features which are not present in the summarised document.

### 6.4 Input word promotion in beam search

To measure the influence of promoting input named entities and nouns during beam search of the abstractive model, we repeated the end-to-end experiment with the input promotions turned off and all other parameters left unchanged. The results of the inference without this feature are shown in table 5. The comparison with results in table 2 shows improvements in all metrics for all compared models when using input promotions.

### 6.5 Assessment of cosine similarity metrics

Cosine similarity between pre-trained sentence embedding measures the semantic relatedness between these sentences irrespective of word overlap. Therefore, this is a promising proxy for comparing quality of generated summaries. As shown in section 6.1, this cosine metric was the only one to predict the best performance of the USEsum model in the end-to-end evaluation. However, depending solely on angle similarities between pairs of embeddings may be misleading, as the same

| Example | Target | System | Inference |
|---|---|---|---|
| 1 | Communal computer available in the lobby | USESum | Great location in the city |
| | | NEUSUM | Offers a flat lounge and free computer room for guests |
| | | BOTTOM-UP | Free Wi-Fi and continental breakfast |
| 2 | Rooftop terrace with fantastic views | USEsum | Rooftop terrace with view over the beach |
| | | NEUSUM | Fitness centre with rooftop terrace |
| | | BOTTOM-UP | Pet-friendly hotel with full kitchens |
| 3 | Funky, modern décor in the heart of Valencia | USESum | Stylish rooms with lovely wooden floors |
| | | NEUSUM | Spacious rooms with extra wooden floors |
| | | BOTTOM-UP | Modern hotel with free breakfast |

Table 3: Example extractive outputs from USEG, NEUSUM, and BOTTOM-UP systems

| System | Grammar % | Semantics % |
|---|---|---|
| USEsum | 60 | **10** |
| NEUSUM | 56 | 8 |
| BOTTOM-UP | **91** | 7.5 |

Table 4: Human evaluation of end-to-end outputs

| | BLEU | ROUGE-L | METOR | Cosine Similarity % |
|---|---|---|---|---|
| BASELINE, No Promotion | 0.0059 | 0.0761 | 0.0265 | 0.4705 |
| NEUSUM, No Promotion | 0.0075 | 0.0838 | 0.0332 | 0.4396 |
| USEsum, No Promotions | 0.0104 | 0.0875 | 0.0337 | 0.4482 |

Table 5: Beam search without promoting source words as output

angle may be calculated for points in very different locations in the semantic embedding space. This results in the assignment of higher cosine similarity scores to the models that excel in averaging predictions. This is demonstrated by the BOTTOM-UP approach outperforming USEsum in cosine similarity as shown in table 2. This is despite BOTTOM-UP generating repetitive predictions. A similar observation can be made in the beamsearch experiment (section 6.4). Results in table 5 show that cosine similarity metrics is useful for comparing similar models (USEsum and NEUSUM) but overly promotes the BASELINE approach which simply averages vectors in a document. Therefore, we can claim that cosine similarity is a valuable metric for assessing the outputs from similar systems.

## 7 Conclusion

We have proposed a novel method for summarising large unstructured documents containing various styles into short uniform styled summaries. The proposed approach outperforms competitive solutions as measured in the USEG-based evaluation.

We found that using sentence embeddings, calculated by Universal Sentence Encoder (Cer et al., 2018), for measuring information gain and similarity is beneficial when processing texts with low levels of word overlap. The results we obtained confirm the semantic meaningfulness and high accuracy of these vectors.

We also showed that by using simple heuristics for adjusting beam scores of word candidates, improves the end-to-end summarisation task with the USEG dataset.

## 8 Future Work

A future extension of the USEsum system would include adaption and evaluation for the *CNN/Daily Mail* newswire corpora. Currently, the decoder of the extractive model is LSTM inspired. We would like to adapt the universal transformer as a decoder in the extractive model. Our motivation is the fact that the decoder based on a transformer used in BERTSUM outperformed the RNN decoder.

Other possibilities include adapting USEsum for other multi-lingual summarisation, beyond English, by using universal language-agnostic sentence level embeddings through implementations such as Facebook's LASER[6].

[6]Facebook LASER — https://code.fb.com/ai-research/laser-multilingual-sentence-embeddings/

## References


Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, Melbourne, Australia. Association for Computing Machinery.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the NAACL Conference*.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*, HLT-NAACL-DUC '03, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv preprint arXiv:1603.08887*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098—4109. Association for Computational Linguistics.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive Summarization of Product Reviews Using Discourse Structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL*.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. *arXiv preprint arXiv:1808.07913*.

Florian Kunneman, Sander Wubben, Emiel Krahmer, and Antal van den Bosch. 2018. Aspect-based summarization of pros and cons in unstructured product reviews. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Chin-Yew Lew. 2004. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017. Focal Loss for Dense Object Detection. In *The IEEE International Conference on Computer Vision (ICCV)*.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL-HLT)*.

Paul Over and James Yen. 2003. An Introduction to DUC-2003: Intrinsic Evaluation of Generic News Text Summarization Systems. In *Proceedings of Document Understanding Workshop 2003*. National Institute of Standards and Technology.

Paul Over and James Yen. 2004. An Introduction to DUC-2004: Intrinsic Evaluation of Generic News Text Summarization Systems. In *Proceedings of Document Understanding Workshop 2004*. National Institute of Standards and Technology.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *eprint arXiv:1705.04304*.

Samuel Pecar. 2018. Towards opinion summarization of customer reviews. In *Proceedings of ACL 2018, Student Research Workshop*, pages 1–8. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.

Evan Sandhaus. 2008. *The New York Times Annotated Corpus Overview*. The New York Times Company, Research and Development, 620 8th Ave 28th Floor New York, NY 10018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *The Thirty-Second AAAI Conference on Aritifical Intelligence (AAAI-18)*, pages 5602–5609.

Lu Yang. 2016. Abstractive summarization for amazon reviews. Technical report, Stanford University.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the ACL Conference*.