Vocabulary Tailored Summary Generation

Kundan Krishna
Adobe Research
kunkrish@adobe.com

Aniket Murhekar
IIT Bombay
aniket1602@gmail.com

Saumitra Sharma
IIT Guwahati
sharmasaumitra15@gmail.com

Balaji Vasan Srinivasan
Adobe Research
balsrini@adobe.com

Abstract

Neural sequence-to-sequence models have been successfully extended for summary generation. However, existing frameworks generate a single summary for a given input and do not tune the summaries towards any additional constraints/preferences. Such a tunable framework is desirable to account for linguistic preferences of the specific audience who will consume the summary. In this paper, we propose a neural framework to generate summaries constrained to vocabulary-defined linguistic preferences of a target audience. The proposed method accounts for the generation context by tuning the summary words at the time of generation. Our evaluations indicate that the proposed approach tunes summaries to the target vocabulary while still maintaining a superior summary quality against a state-of-the-art word embedding based lexical substitution algorithm, suggesting the feasibility of the proposed approach. We demonstrate two applications of the proposed approach - to generate understandable summaries with simpler words, and readable summaries with shorter words.

1 Introduction

Automatic text summarization (Nenkova and McKeown, 2011) is the task of generating summaries of an input document while retaining the important points. These summaries are used for presenting the most relevant and important information in a long text in a succinct form. They are useful in places where a quick consumption of the information in a long article is preferred. Earlier works in summarization select sentences/textual units from the input article and put them together into an "extractive" summary. However, humans summarize an article by understanding the content and paraphrasing the understood content into the desired summary. Therefore, extractive summarization is unable to produce "human-like" summaries. This has led to efforts towards "abstractive" summarization which paraphrases summaries the input article. Several models have been proposed, with the most recent ones based on neural networks.

Often, it is desirable to tune the summaries to the linguistic preferences of the readers. For example, a medical report may contain a lot of technical jargon beyond the understanding of the common population. When such a report is consumed by a patient, it makes sense to use lesser jargon to suit a patient's knowledge level. Similarly, while reading articles, teenagers would prefer more informal and trendy words, while older people might like a more formal vocabulary. Summaries which are tuned to such "linguistic" preferences of the target population segment are likely to appeal better and catch their attention.

A standard approach to incorporate vocabulary tuning would be to post-process a generated summary to achieve the desired goal by replacing a few words (e.g. replacing words with their simpler alternatives). However, such an approach might not preserve the context and hence can result in a complete change in the meaning of the content. Consider the sentence *"The baseball **pitcher** was seen with a **pitcher** of beer in his hand."* The word *pitcher* means different things in its two occurrences here. The sentence can be rephrased as *"The baseball **player** was seen with a **jug** of beer in his hand."* Since *pitcher* can mean both *player* and *jug* in two independent senses, it is not easy to decide the right replacement without looking

Proceedings of the 27th International Conference on Computational Linguistics, pages 795–805
Santa Fe, New Mexico, USA, August 20-26, 2018.

at the respective context. Post-processing based approaches lose the contextual information from the source article once the summary has been constructed. This is because the algorithm only sees the short snippet of text (summary), and not the large source article. This is a lost opportunity to utilize all that contextual information to make better word substitutions.

In this paper, we propose a neural network based summary generator that generates summaries imposing the desired vocabulary preferences while also maintaining the context and meaning of the content. Neural network based summarizers encode the entire source article, and generate the summary word-by-word. They are trained to predict the next word in a summary given the words in the summary generated so far, as well as the encoded article. This will allow the network to better judge which words will fit in context at any position in the sentence. In our proposed approach, we modify the generation probability of the next word in accordance with the vocabulary-based preferences. A key advantage of the proposed approach is that it does not require re-training a neural network, and relies on modifying the summary generation procedure on a pre-trained network.

2 Related Work

Traditional methods for summarization (Nenkova and McKeown, 2011) extract key sentences from the source text to construct the summary based on various features like descriptiveness of words, word frequencies, etc. Early attempts at abstractive summarization created summary sentences based on templates (Wang and Cardie, 2013; Genest and Lapalme, 2011) or used ILP-based sentence compression to collect parts from various sentences to generate the summary (Filippova, 2010; Berg-Kirkpatrick et al., 2011; Banerjee et al., 2015).

With the advent of deep sequence to sequence networks (Sutskever et al., 2014), attention based models have been proposed for summarizing long sentences (Rush et al., 2015; Chopra et al., 2016). Gulcehre et al. (2016) incorporated the ability to copy out-of-vocabulary words from the article to incorporate rarely seen words like names in the generated text. Tu et al. (2016) included the concept of coverage, to prevent the models from repeating the same phrases while generating a sentence. See et al. (2017) proposed a summarization model which incorporates these improvements, and also learns to switch between generating new words and copying words from the source article. We use this summarization framework as the starting point for our work. However, none of the existing works attempt to tune the summaries to suit preferences of a reader.

One naive way to solve this problem could be to impose the preferences after the summary generation as a post-processing step. As we will show later, this can result in out-of-context replacements while tuning. The words can be substituted based on a standard thesaurus (Bott et al., 2012) using one of the synonyms of the target word. However, since a word can be used in multiple senses, not all of its synonyms can be used in its place, and therefore such an approach is prone to errors. We address these challenges in the proposed approach by optimizing for the vocabulary preferences at the time of summary generation by looking for potential replacements in the synonyms weighted by their contribution to the context (computed from the attention models). As we will show later, such an approach generates better quality summaries and reduces substitution errors.

Lexical substitution deals with deciding textual substitutions that will preserve the meaning and grammatical correctness of the sentence by modeling the overall sentence context and using it for word substitutions. Early methods used co-occurrence statistics of the possible substitutions and the context words to predict whether a substitution is valid in a given context (Szarvas et al., 2013).

Melamud et al. (2015) use the proximity of words in an embedding space to measure the appropriateness of a candidate substitution to replace the target word. They use the embeddings of the dependency relations (De Marneffe and Manning, 2008) of the target word in the same embedding space called 'context embeddings'. The cosine similarity between the embedding of a candidate substitution with embeddings of these dependency relations along with the target was shown to improve substitution performance. Roller and Erk (2016) extended this work further by incorporating a linear transformation of the context embeddings and learning the parameters of the transformation to rank possible substitutions.

These methods work on the hypothesis that words closer to the target word in the embedding space are

its viable replacements. We believe that this hypothesis might not always hold. While it is true that the proximity of word embeddings of two words implies their usage in similar contexts (similar neighboring words or dependency relations), two different words having opposite meanings can also occur nearby in the word embedding space. For example, *good* and *bad* have very close embeddings in the space trained on the Google News corpus[1] by Mikolov et al. (2013), because both are adjectives and used around similar contexts. In this embedding space, the cosine similarity between *good* and *bad* is 0.72, whereas similarity between *good* and *wonderful* is 0.57. However, replacing good with bad will certainly change the meaning of the sentence and despite having a lower similarity, *wonderful* is the better substitute in most cases. Our method does not suffer from such incorrect substitutions because we couple the information from a thesaurus with the contextual information from the neural decoder to generate the summary by picking the appropriate words. We show that compared to contextual word vectors, the neural decoder is able to capture the context better and so our method generates better summaries.

Another related line of work is **text adaptation** which deals with modification of the textual content to suit the needs of a particular audience segment. Text simplification (Paetzold and Specia, 2015; Paetzold and Specia, 2016) is a popular variant of text adaptation where the objective is to modify text to have simpler words so that it is easier to comprehend. Linguistic personalization is another variant of the problem which looks at modifying messages to suit a target segment's linguistic style (Roy et al., 2015). However, all these approaches adapt the text as a post-processing task, and hence do not account for the context with which the text was generated. Our proposed approach is generic and can be extended to address these variants of text adaptation while accounting for the context of generation. In particular, we show the application to the tasks of text simplification and text readability enhancement.

3 Summary Generation with Vocabulary Tuning (VoTing)

Given an input text article as a sequence of n tokens $A = a_1, a_2, ..., a_n$, a vocabulary $V = \{w_1, w_2, ..., w_k\}$ of words with scores indicating the preference of each word given by $q : V \rightarrow \mathbb{R}^+$, the objective is to generate a summary as a sequence of tokens $S = s_1, s_2, ..., s_m$ tuned to the preferences indicated by the vocabulary while preserving the contextual sense.

We extend the **pointer generator network** by See et al. (2017) to generate the summary in a word-by-word fashion. At each step, the algorithm runs a trained decoder neural network to output the probability of each word $w \in V$ being the next generated word. This generation probability of any word is also an indicator of its contextual appropriateness in the current generation.

Our primary contribution in this paper is a **modified decoding algorithm** to incorporate vocabulary preferences. At each decoding step of the trained neural network, instead of adding the word w having the highest generation probability to the summary, we tune it by replacing it with a better preferred word that is also contextually appropriate. We take all synonyms of w and score their contextual appropriateness based on their generation probabilities. We combine the contextual appropriateness with the "vocabulary" scores of the synonyms based on the vocabulary metric q (e.g. simplicity) to select the contextually best candidate that is preferred in the vocabulary, and append it to the summary. Iteratively repeating this builds the complete summary.

3.1 Pointer Generator Network

For the sake of completeness and introducing the notations, we first give an overview of the pointer generator architecture (See et al., 2017) before introducing our vocabulary tuning approach in Section 3.2. The pointer generator network consists of an encoder and a decoder, both based on LSTM architecture. Given an input article, the encoder takes the word embedding vectors of the source text $A = a_1 a_2 ... a_n$ and computes a sequence of encoder hidden states $h_1, h_2, ..h_n$. The final hidden state is passed to a decoder. The decoder computes a hidden state s_t at each decoding time step, and an attention distribution a^t is over all words in the source text,

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{att});\ a^t = \text{softmax}(e^t) \quad (1)$$

[1]Pretrained word embeddings available at `https://code.google.com/archive/p/word2vec/`

where v, W_h, W_s and b_{att} are trained model parameters. The attention model is a probability distribution over the words in the source text, which aids the decoder in generating the next word in the summary using words with higher attention. The context vector h_t^* is a weighted sum of the encoder hidden states and is used to determine the next word that is to be generated.

$$h_t^* = \sum_{i=1}^{n} a_i^t h_i, \tag{2}$$

At each decoding time step, the decoder uses the last word y_t in the summary generated so far and computes a scalar p_{gen} denoting the probability of the network generating a new word from the vocabulary.

$$p_{gen} = \sigma(w_h^T h_t^* + w_s^T s_t + w_y^T y_t + b_{gen}) \tag{3}$$

where w_h, w_s, w_y, b_{gen} are trained vectors. The network probabilistically decides based on p_{gen}, whether to generate a new word from the vocabulary or copy a word from the source text using the attention distribution. For each word w in the vocabulary, the model calculates $P_{vocab}(w)$, the probability of the word getting newly generated next. P_{vocab} is calculated by passing a concatenation s_t and h_t^* through a linear transformation with softmax activation. On the other hand, for each word w' in the input article, its total attention received yields its probability of being copied. The total probability of w being the next word generated in the summary, denoted by $\mathbf{p}$ is given by,

$$\mathbf{p}(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \tag{4}$$

The second term allows the framework to choose a word to copy from the input text using the attention distribution. For our experiments, we utilized the model trained using back-propagation and the Adagrad gradient descent algorithm (Duchi et al., 2011).

3.2 Vocabulary Tuning (VoTing)

We assume that there exists a scoring function $q(w)$ that computes a quality/preference score of a word w in the target vocabulary. The preference can be along different criteria like simplicity, readability, commonality etc. There have been several approaches explored (Paetzold and Specia, 2015) to compute such a score for a target corpus/vocabulary, e.g. normalized word counts in the target vocabulary. Our primary contribution is a method to integrate such preference scores with the generation process for tuning the generated summary.

We encourage the decoder to pick a different word w' in place of w if it has a higher preference score and similar contextual appropriateness ($\mathbf{p}()$). To preserve the meaning in the generation, we restrict the possible replacements to synonyms of w, $w' \in Syn(w)$, where $Syn(w)$ represents the set of synonyms of w. Since the replacement is done to tune the summary to the target vocabulary, it implicitly attempts to maximize the aggregated quality/preference score of the generated summary. We therefore, define the probability of replacement $p(w'|w)$ to be non-zero only when the new word(w') has a higher quality/preference score than the old one(w). We calculate $p(w'|w)$ for each pair of words (w', w) in the vocabulary which is given by,

$$p(w'|w) = \begin{cases} \frac{q(w')}{N(w)} & \text{if } q(w') \geq q(w) \text{ and } w' \in Syn(w) \\ 0 & \text{otherwise} \end{cases} ; \text{ where } N(w) = \sum_{q(w'') \geq q(w), w'' \in Syn(w)} q(w'')$$

$p(w'|w)$ can be seen as the replacement affinities for a word w with respect to other words in the vocabulary. Whenever the decoder adds a token w (the token with the highest generation probability from the network) to the summary, we calculate,

$$w_{tuned} = \underset{w':p(w'|w)>0}{\arg\max} \hat{\mathbf{p}}(w') p(w'|w) \tag{5}$$

$$\hat{\mathbf{p}}(w') = \frac{e^{(\ln \mathbf{p}(w'))/r}}{\sum_{\bar{w}:p(\bar{w}|w)>0} e^{(\ln \mathbf{p}(\bar{w}))/r}} \tag{6}$$

where $\mathbf{p}$ is the distribution from the network in the latest time step, which contains the generation probabilities for each word in the vocabulary. This is an indicator of the current contextual appropriateness of the words in the vocabulary. The vocabulary preferences from $p(w'|w)$ is thus combined with the contextual appropriateness from $\hat{\mathbf{p}}(w')$ (a function of $\mathbf{p}$). The token w_{tuned} thus obtained from Eq. 5 is added to the tuned summary by replacing w. Eq. 6 (inspired by the softmax activation function with temperature often used in reinforcement learning (Sutton and Barto, 1998)) includes a replacement strength parameter r that can be used to tune the replacement levels for the algorithm. The value of r is always kept positive.

When r is close to 0, the distribution of $\hat{\mathbf{p}}(w')$ is more peaked and the value of $\hat{\mathbf{p}}(w')$ is almost 1 for the w' having highest $\mathbf{p}(w')$ and almost 0 for others. Hence $w_{tuned} = \arg\max_{w'} \mathbf{p}(w')$, and there are no replacements to tune for vocabulary. As r increases to 1, we see more replacements. When r goes much higher than 1, $\hat{\mathbf{p}}(w')$ tend to be almost same across all w'. Hence, the output would depend more on the target vocabulary preferences $p(w'|w)$, leading to more aggressive replacements at the cost of contextual appropriateness.

The proposed decoder has a better understanding of context because of the awareness of past words produced in the summary. Besides, attention based decoding has been shown to generate new words while preserving context, like generating the word *beat* by paying attention to words like *victorious* and *win* from the source text of an article about a football match (See et al., 2017). This suggests that there are high probabilities of generation for novel synonyms which can actually be used in the summary while preserving context.

As we will show later in our experiments in Section 4.2, the source article itself might have more than one word appropriate for a given context, which can be used by the decoder. For example, an article about "crime" can have both words - *inexplicable* and *mysterious*. Since generation probabilities of the pointer generator network are influenced partly by its tendency of copying words from source text, these words have a high probability of generation($\mathbf{p}()$). Now if our algorithm has to choose an alternative word for *inexplicable*, it is more likely to generate *mysterious* (if it is better suited for the target vocabulary), because of its higher generation probability than other synonyms of *inexplicable* which are not in the text and may or may not be usable in the given context.

4 Experiments

The proposed approach can be used in applications where the audience's linguistic preference can be quantified. Here, we evaluate the framework on two such applications: enhancing **text simplicity** and enhancing **text readability** of generated summaries. In the former experiment, our objective is to generate a "simple" summary that contains more commonly used words. In the latter, we adapt the summary to use shorter words thereby making it more readable. In both applications, we test the proposed framework against several baselines that are described below.

Pointer-Generator Summary (PGS): These are the summaries generated by the vanilla pointer-generator network (See et al., 2017) without any optimization for vocabulary. Note that the proposed algorithm is aimed at producing summaries with comparable quality to this vanilla generator and better tuned to the vocabulary.

Non-Contextual Post-processed Summary (NCP): Here, we replace a given word with its synonym which has the highest score. All replacements are carried out after the summaries have been generated by the network. Note that this does not consider the context for the summary generation.

Contextual Word Embedding (CWE): This method is based on the hypothesis that cosine similarities in the word/dependency embedding space capture the extent to which a word can contextually replace another as proposed by Melamud et al. (2015). Given a target word t to be replaced in a sentence, $\mathbf{p}(w'|t)$ defines a measure of the appropriateness of the word w' replacing t. If t has m dependency relations $r_1, r_2, ...r_m$ with words $w_1, w_2, .., w_m$, then $\mathbf{p}(w'|t)$ is given by,

$$\mathbf{p}(w'|t) = \frac{1}{2m}(m\langle v(w'), v(t)\rangle + \sum_{i=1}^{m}\langle v(w'), v(r_i, w_i)\rangle) \quad (7)$$

where, $v(w)$ is the embedding vector for word w and $v(r_i, w_i)$ is the embedding vector for a dependency in the same embedding space. We extend this towards our problem by replacing each word w by w_{CWE} in the generated summary S based on,

$$w_{\text{CWE}} = \underset{w':p(w'|w)>0}{\arg\max}\ \hat{\mathbf{p}}(w'|w)p(w'|w) \tag{8}$$

We define $\hat{\mathbf{p}}(w'|w)$ similar to our formulation in Eq. 6 as,

$$\hat{\mathbf{p}}(w'|w) = \frac{e^{\mathbf{p}(w'|w)/r}}{\sum\limits_{\bar{w}:p(\bar{w}|w)>0} e^{\mathbf{p}(\bar{w}|w)/r}} \tag{9}$$

where r is the replacement strength parameter. To enhance the overall quality of replacements and for unbiased comparisons, all compared approaches were set to not replace stopwords.

4.1 Dataset & Evaluation Metric

We use the CNN/Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016) which consists of $312,084$ news articles from the CNN and Daily Mail news websites, together with multi-line human-written summaries. The dataset consists of $287,226$ article-summary training pairs, $13,368$ validation pairs and $11,490$ test pairs.

Besides measuring the degree to which the vocabulary has been tuned, we also use **ROUGE** scores (Lin, 2004) to calculate the degree of similarity between the algorithmically generated summary and a human generated summary in terms of overlap of unigrams (ROUGE-1), bi-grams (ROUGE-2) and longest common subsequence (ROUGE-L).

4.2 Simplified Summary Generation

Our first experiment focuses on simplified summary generation. Simplification aims at rewording given text to make it simpler to understand for an audience. Existing works in simplification (Paetzold and Specia, 2015; Paetzold and Specia, 2016) break down the problem into a pipeline with multiple steps: complex word identification, substitution generation, substitution selection and substitution ranking. The generation of possible substitutions can be done in many ways (Paetzold and Specia, 2015), some of which leverage dictionaries (Yamamoto, 2013), while others leverage learned substitutions from a parallel corpus (Horn et al., 2014). However, Paetzold and Specia (2016) found that using nearest neighbors in word2vec embedding space leads to better performance in substitution generation. The substitution selection part is also responsible for ensuring the contextual appropriateness of the new word. The ranking of words is typically based on the frequency of words in a standard simple corpus. The hypothesis here, is that amongst the words with the same meaning, the ones which are used frequently are simpler.

Following existing works, we set the score $q(w)$ of a word w to be its frequency in the SUBLTEX corpus (Brysbaert and New, 2009) and tune the summary generation. We measure the simplicity of a summary S based on the simplicity score defined as,

$$\text{Simplicity}(S) = \frac{1}{m}\sum_{i=1}^{m}\frac{f(s_i)}{1000}, \tag{10}$$

where $f(s_i)$ is the frequency of the i^{th} word of the summary in the SUBTLEX corpus.

Table 1(a) shows the performance of various methods across different metrics. For CWE and Voting, we tune r to yield a comparable simplicity score and report the other metrics for this setting.The NCP method achieves the highest simplicity score at the cost of the summary quality as shown by low ROUGE scores. A higher number of replacements will decrease the ROUGE scores if the newly introduced replacements are out of context and therefore unlikely to be in the ground truth summary. We observed that NCP replaces 18.429 words per summary (across the 11490 test set) in this experiment where the average summary length is 57.436 words. In contrast, CWE replaces 2.082 words and VoTing replaces

Table 1: Performance of the proposed approach against existing baselines

(a) Simplified Summary

Metric	PGS	NCP	VoTing	CWE
rouge-1 F-score	0.3880	0.2940	0.3790	0.3771
rouge-2 F-score	0.1679	0.0799	0.1563	0.1552
rouge-L F-score	0.3569	0.2706	0.3482	0.3466
Simplicity score	9.67	28.05	12.51	12.35

(b) Readable Summary

Metric	PGS	NCP	VoTing	CWE
rouge-1 F-score	0.3880	0.2724	0.3810	0.3802
rouge-2 F-score	0.1679	0.0693	0.1593	0.1593
rouge-L F-score	0.3569	0.2525	0.3504	0.3498
Reading ease	59.23	80.90	64.15	64.12

2.231 words per summary. Despite VoTing replacing more words (around 2000 more words in total), it manages to score higher on the ROUGE scores suggesting that the new words introduced still keep the summary closer to ground truth while simultaneously increasing the desired vocabulary score (which is the simplicity in this case). Table 2 shows the choices made by our proposed approach towards summary generation using simpler words from the source article.

Table 2: Simplified summaries where our method picks up simpler words, highlighted in boldface, from the source article. Baseline summaries used the words given in parantheses instead

Article: hong kong (cnn) six people were **hurt** after an explosion (...) *Summary:* (...) five out of six people were **hurt** (injured) by broken glass (...)
Article: (cnn) five americans who were monitored for three weeks at an omaha , nebraska , hospital after being exposed to ebola in west africa have been **released**, (...) *Summary:* one of the five had a heart-related issue on saturday and has been **released** (discharged).(...)
Article: (...)"it is shameful that so many states around the **world** are essentially playing with people's lives (...) *Summary:* (...) china is also mentioned , as having used the death penalty as a punitive measure across the **world** (globe).
Article: (...) but corliss was not afraid to puncture hype around **big** movies he found overrated, including "titanic" (...) *Summary:* richard corliss died a week after suffering a **big** (major) stroke. (...)

Table 3(a) shows the summaries generated on one of the articles by these methods. We can see that NCP over-replaces words leading to loss of meaning. For example, it replaces the word *march* (which refers to a month here) by *move* since move is a valid synonym of one of the senses of the replaced word (e.g. *"The army contingent marched towards the fortress."*). VoTing makes fewer replacements which seem to be in context, like replacing *mom* with *ma* and *reversed* with *turned*. CWE replaces *mom* by *grandmother* which leads to factual incorrectness.

Our formulation in VoTing and CWE allows to control the strength of replacement in the algorithm (Eq. 6 and Eq. 9). Higher strength increases the simplicity score but at the cost of reduced ROUGE. To better compare VoTing and CWE, we must look at the quality of summaries for different levels of simplicity desired in the output. We show these in Figure 1(a). It is observed that VoTing is able to achieve higher ROUGE for any given level of simplicity.

4.3 Readable summary generation

In our next experiment, we focus on readable summary generation. To make text more readable, it is advisable to use words with fewer syllables (Kincaid et al., 1975). Kincaid et al. (1975) define the Flesch reading ease to quantify readability. Text which scores high on the Flesch reading ease can be understood more easily by students of lower grade levels (Flesch, 1979). The Flesch reading ease of a summary S is given by,

$$206.835 - 1.015\frac{\text{total words}}{\text{total sentences}} - 84.6\frac{\text{total syllables}}{\text{total words}}$$

To generate more readable summaries, we run VoTing with higher scores given to shorter words, as they are likely to have fewer syllables. Here, $q(w)$ is set to be the inverse of the length of the word w. This encourages the algorithm to use shorter words while generating summaries.

Here again, we tune r for VoTing and CWE to yield a comparable reading ease and report the other metrics for this setting. Table 1(b) shows the performance of different algorithms. Our method achieves comparable ROUGE to CWE, indicating that we achieve the target without compromising on the quality.

Table 3(b) shows some sample outputs for an article. NCP replaces most words with their shortest synonyms leading to complete loss of meaning. CWE makes fewer substitutions which are more appro-

Table 3: Summaries generated by different methods on a sample article. Changed words from the baseline are highlighted in boldface

(a) Simple Summary	(b) Readble Summary
Article: Facebook has admitted it made a mistake when a photo of an Alabama boy who was born without a nose was removed from the social media website because it was deemed to be too controversial. The photo of Timothy Eli Thompson that was removed when a pro-life group posted an ad about the infant's story have since been reinstated. (...)	*Article:* A Paratrooper who braved heavy Taliban fire to rescue a wounded comrade received the Victoria Cross from the Queen yesterday. She told Lance Corporal Joshua Leakey: I dont get to give this one out very often. Did you ever imagine youd be standing here? Well done. But in fact the 27-year-old is the second member of his family to receive the highest military decoration for valour a cousin was given the honour 70 years ago (...)
Baseline summary: timothy eli thompson was born without a nose in march in alabama and his mom put his photo on facebook .facebook reversed its decision after public protest and admitted it made a mistake .facebook reversed its decision after public protest .	*Baseline summary:* the 27-year-old is the second member of his family to receive the highest military decoration for valour .it is just the sixth time the queen has given a vc to a living british recipient during the afghanistan campaign but the other two awards were made posthumously .in 2013 he braved heavy gunfire from 20 taliban insurgents in helmand to rush to the aid of a wounded us marine .
VoTing: timothy eli thompson was born without a nose in march in alabama and his **ma** put his photo on facebook . facebook **turned** its decision after public protest and admitted it made a mistake . facebook reversed its decision after public protest .	*VoTing:* the 27-year-old is the second member of his family to **get** the **top** military decoration for valour . it is just the sixth time the queen has given a vc to a living british recipient during the afghanistan campaign but the other two awards were made posthumously . in 2013 he braved heavy gunfire from 20 taliban insurgents in helmand to rush to the aid of a wounded us marine .
CWE: timothy eli thompson was born without a nose in march in alabama and his **grandmother** put his photo on facebook . facebook reversed its decision after public protest and admitted it made a mistake . facebook reversed its decision after public protest .	*CWE:* the 27-year-old is the second member of his family to **get** the highest military decoration for valour . it is just the sixth time the queen has given a vc to a living british recipient during the afghanistan campaign but the other two awards were made posthumously . in 2013 he braved heavy gunfire from 20 taliban insurgents in helmand to rush to the aid of a wounded us marine .
NCP: timothy eli thompson was born **out** a nose in **move** in alabama and his **ma** put his **picture** on facebook . facebook **turned** its **end** after **open question** and **take** it made a **fault** . facebook **turned** its **end** after **open question** .	*NCP:* the 27-year-old is the **twin cut** of his **clan** to **cop** the **top army garnish** for valour . it is just the sixth **go** the **ruler** has **apt** a vc to a **warm** british **heir** during the afghanistan **push** but the other **dos gift** were made posthumously . in 2013 he **firm fat salvo** from 20 taliban **radical** in helmand to **flux** to the aid of a **hurt** us **sea** .

priate like replacing *receive* by *get*. VoTing makes the same replacement and also replaces *highest* by *top* - suggesting that our approach performs more tuning without compromising on the overall quality.

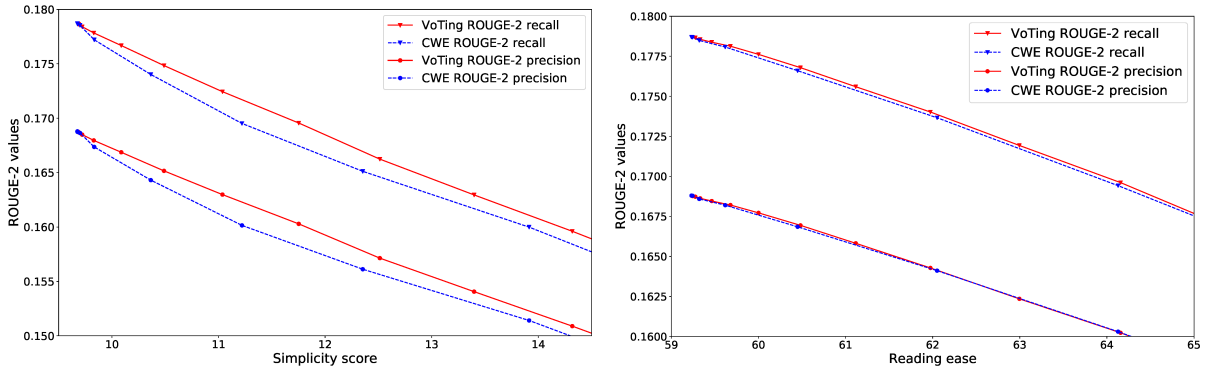
Varying the replacement strength parameter, we find that VoTing again has higher ROUGE-2 scores across different levels of reading ease. This can be seen in Figure 1(b) indicating better contextual tuning by the proposed approach across different reading ease.

4.4 Human evaluation of contextual appropriateness of VoTing

While the level of improvement in the vocabulary tuning achieved in a summary can be measured by using various scores, the contextual appropriateness of the new words added is better judged by humans. We, therefore, conducted a survey, where each annotator was shown the ground truth summary generated by PGS along with the tuned summaries from the three methods - NCP, CWE and VoTing. The annotators were asked to rank the outputs of the three methods according to the extent to which it preserves the meaning of the original summary generated by PGS. The three tuned summaries were shown in random order to remove any positional bias. We used the summaries from 20 randomly chosen articles from the test set for the survey. Each set of summaries was annotated by 4 or 5 different annotators. We had a total of 90 human annotated rankings.

We used the Condorcet fusion (Montague and Aslam, 2002) to aggregate the rankings, which looks at pairwise comparisions between the methods. The results are shown in Table 4, which indicates that VoTing performs the best, beating CWE in 61.11% of responses, and NCP clearly performs the worst. The Krippendorff's alpha (Krippendorff, 2011) for inter-annotator agreement was 0.84 indicating high inter-annotator agreement.

After establishing the comparative superiority of VoTing, we proceeded to objectively analyze the degree to which these three methods preserved the meaning of the original summary. We ran another



(a) Simple Summary

(b) Readable Summary

Figure 1: CWE vs Voting for different simplicity and reading ease levels. ROUGE-2 precision and recall are shown for different levels of tuning achieved.

Table 4: Pairwise comparison of human rankings of different methods. Each row signifies the fraction of times the corresponding method was ranked higher than the method corresponding to the column.

	CWE	**VoTing**	**NCP**
CWE	-	0.3889	0.9667
VoTing	0.6111	-	0.9778
NCP	0.0333	0.0222	-

human evaluation where we showed human raters the summaries from the three methods in random order and asked them to rate the three on a scale of 1 to 5 according to the descriptions given in Table 5, on the extent to which they preserve the meaning of the PGS generated summary. VoTing received an average rating of 3.47 against 3.36 for CWE and 1.83 for NCP, further confirming the contextual appropriateness of the proposed tuning.

Table 5: Description shown to human annotators for ratings on the contextual appropriateness scale

Rating value	**Description**
5	Perfectly captures the original meaning
4	Mostly preserves the meaning
3	Understandably close to the original meaning
2	Changes the meaning by a little bit
1	Completely changes the meaning

5 Conclusions and Future work

We proposed a novel approach to generate summaries of articles while incorporating vocabulary preferences. We showed the application of our algorithm to text simplification and text readability enhancement. We showed that tuning the vocabulary during summary generation leads to fewer out-of-context replacements than post-processing a generated summary. Our findings also suggest that LSTM-based decoders of pointer-generator networks are capable of preserving the local context better than word embeddings trained on vast corpora.

Our current work is limited to replacing words with better synonyms. However, introduction of new words can benefit tuning the generation towards a specific aspect or tone. For example, "Pass me that plate." can be changed to "*Please* pass me that plate." to make it sound more formal. Rephrasing a sequence of words instead of replacing one word at a time or changing the structure of a sentence are other ways to make it suit a target audience's preference. The ability of LSTM based decoders to carry out such transformations is a subject for further explorations.

References

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ilp based multi-sentence compression. In *International Joint Conference on Artificial Intelligence*.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 481–490. Association for Computational Linguistics.

Stefan Bott, Luz Rello, Biljana Drndarevic, and Horacio Saggion. 2012. Can spanish be simpler? lexsis: Lexical simplification for spanish. *International Conference on Computational Linguistics*, pages 357–374.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*.

Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 93–98.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *International Conference on Computational Linguistics (COLING)*. Association for Computational Linguistics.

Rudolf Franz Flesch. 1979. *How to write plain English: A book for lawyers and consumers*. Harpercollins.

Pierre-Etienne Genest and Guy Lapalme. 2011. Framework for abstractive summarization using text-to-text generation. In *Workshop on Monolingual Text-To-Text Generation*. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Karl Moritz Hermann, Tomas Kocisky, 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*.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. *ScholarlyCommons*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*. Barcelona, Spain.

Oren Melamud, Omer Levy, Ido Dagan, and Israel Ramat-Gan. 2015. A simple word embedding model for lexical substitution. In *VS@ HLT-NAACL*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*.

Mark Montague and Javed A Aslam. 2002. Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 538–548. ACM.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL*.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*.

Gustavo Henrique Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. In *ACL (System Demonstrations)*.

Gustavo Henrique Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *30th AAAI Conference (ACL)on Artificial Intelligence*.

Stephen Roller and Katrin Erk. 2016. Pic a different word: A simple model for lexical substitution in context. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Rishiraj Saha Roy, Aishwarya Padmakumar, Guna Prasaad Jeganathan, and Ponnurangam Kumaraguru. 2015. Automated linguistic personalization of targeted marketing messages mining user-generated text on social media. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Conference on Empirical Methods in Natural Language Processing*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

György Szarvas, Chris Biemann, Iryna Gurevych, et al. 2013. Supervised all-words lexical substitution using delexicalized features. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

T Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the Twenty-Fifth Conference on Computational Linguistics and Speech Processing (ROCLING 2013)*.