Using contextual and cross-lingual word embeddings to improve variety in template-based NLG for automated journalism

Miia Rämö
University of Helsinki
Department of Computer Science
miia.ramo@helsinki.fi

Leo Leppänen
University of Helsinki
Department of Computer Science
leo.leppanen@helsinki.fi

Abstract

In this work, we describe our efforts in improving the variety of language generated from a rule-based NLG system for automated journalism. We present two approaches: one based on inserting completely new words into sentences generated from templates, and another based on replacing words with synonyms. Our initial results from a human evaluation conducted in English indicate that these approaches successfully improve the variety of the language without significantly modifying sentence meaning. We also present variations of the methods applicable to low-resource languages, simulated here using Finnish, where cross-lingual aligned embeddings are harnessed to make use of linguistic resources in a high-resource language. A human evaluation indicates that while proposed methods show potential in the low-resource case, additional work is needed to improve their performance.

1 Introduction

The use of automation to help journalists in news production is of great interest to many newsrooms across the world (Fanta, 2017; Sirén-Heikel et al., 2019). *Natural Language Generation* (NLG) methods have previously been employed, for example, to produce soccer reports (Chen and Mooney, 2008), financial reports (Plachouras et al., 2016) and weather forecasts (Goldberg et al., 1994). Such 'automated journalism' (Carlson, 2015; Graefe, 2016) or 'news automation' (Sirén-Heikel et al., 2019) imposes restrictions on system aspects such as transparency, accuracy, modifiability, transferability and output's fluency (Leppänen et al., 2017). Likely as a consequence of these requirements, news industry applications of NLG have traditionally employed the 'classical' rule-based approaches to NLG, rather than the more recent neural methods increasingly seen in recent academic literature (Sirén-Heikel et al., 2019). A major downside of these rule-based systems, however, is that their output often lacks variety. Adding variety by increasing the amount of templates is possible, but this would significantly increase the cost of system creation and limits reuse potential. As users of automated journalism already find the difficulty of reuse limiting (Linden, 2017), this is not a sustainable solution.

In this paper, we extend a modular rule-based NLG system – used for automated journalism in the domain of statistical news – with a dedicated component for varying the produced language in a controlled manner. The proposed extension enables two methods of inducing further variation: in *insertion*, new words are introduced into the generated text, whereas in *replacement* certain words in the original sentence are replaced with synonyms. To accomplish these tasks, we employ a combination of traditional language resources (e.g. synonym dictionaries) as well as recent neural processing models (i.e. word embeddings). These resources complement each other, enabling us to harness the power of statistical NLP tools while retaining control via the classical linguistic resources. We also experiment with using these methods in the context of a *low-resource* language which lacks linguistic resources such as synonym dictionaries. For this case, we propose to use cross-lingual aligned word embeddings to utilize a high-resource language's resources even within said low-resource language.

In the next section, we briefly describe some related previous works and further motivate our approach. Section 3 describes our proposed variation induction methods for both the high-resource and the low-resource contexts. Sections 4 and 5, respectively, introduce our human evaluation method and the results obtained. Sections 6 provides some additional thoughts on these results, while Section 7 concludes the paper.

Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation, pages 62–70
April 19, 2021

2 Background

Natural language generation has been associated with news production from the early years of the field, with some of the earliest industry applications of the NLG methods being in the domain of weather report production (Goldberg et al., 1994). Interest in applying NLG to news production has only increased since, with many media houses experimenting with the technology (Fanta, 2017; Sirén-Heikel et al., 2019). Still, adoption of automated journalism methods has been slow. According to news media insiders, rule-based, classical, NLG system such as those described by Reiter and Dale (2000), are costly to create and difficult to reuse (Linden, 2017). At the same time, even the most recent neural (end-to-end) approaches to NLG are not fit for customer needs as they limit the ability to "customise, configure, and control the content and terminology" (Reiter, 2019). Another major problem is the fact they suffer from a form of overfitting known as 'hallucination', where ungrounded output text is produced. This is catastrophic in automated journalism.

Concurrently with works on improved neural NLG methods, others have investigated increasingly modular rule-based approaches with the intent of addressing the reusability problem described by Linden (2017). For example, Leppänen et al. (2017) describe a modular rule-based system for automated journalism that seeks to separate text domain specific processing from language specific processing to allow for easier transfer of the system to new text domains. While such rule-based approaches produce output that is grammatically and factually correct (Gatt and Krahmer, 2017), they often suffer from a lack of variety in language. This is especially true for systems that are based on some type of *templates*, or fragmentary language resources that are combined to form larger segments of text and into which content dependent on system input is embedded. Using such templates (or hand-crafted grammars) is costly, especially when a large number is required for varied output.

As template (or grammar) production can be costly, automated variation induction methods that could be integrated into rule-based systems are very interesting. One trivial approach to inducing variation would be to employ a synonym dictionary, such as is available in WordNet (Miller, 1995), to replace words within the generated text with their synonyms. This approach, however, suffers from some major problems. First, simply looking up all synonyms for all meanings of a token is not feasible due to polysemy and homonymy. At the same time, incorporating knowledge of which semantic meaning of a token is correct in each case significantly slows down template and grammar generation. Furthermore, even within a certain semantic meaning, the various (near) synonyms might not be equally suitable for a given context. Finally, such linguistic resources are not available for many low-resource languages.

An alternative approach, more suited to generation within medium and low-resource languages where there are no available synonym dictionaries, but large text corpora can be collected, would be to use word embeddings (E.g. Rumelhart et al., 1986; Bengio et al., 2003; Mikolov et al., 2013) to identify words that are semantically close to the words in the template. This approach, however, suffers from the fact that both synonyms and antonyms of a word reside close to it in the word embedding space. While potential solutions have been proposed (E.g. Nguyen et al., 2016), they are not foolproof.

3 Variety Induction Algorithms

As described above, naïve methods based on either classical linguistic resources or word embeddings alone are not suitable for variation induction. To this end, we are interested in identifying a simple variety induction method that combines the positive sides of both the classical linguistic resources (such as synonym dictionaries) with those of statistical resources such as word embeddings. Optimally, the method should also function for a wide variety of languages, including low-resource languages where costly resources such as comprehensive synonym dictionaries are not readily available.

In this work, we introduce variety into the generated language using two distinct methods: by introducing completely new words into sentences, and by replacing existing words. We will use the terms *insertion* and *replacement* to distinguish between the two approaches, respectively.

3.1 Introducing Variety with Insertion

In our insertion method, new words are introduced to sentences at locations where placeholder tokens are defined in templates. We use a combination of a part-of-speech (POS) tagger and a contextual language model to control the process. A simplified

Algorithm 1 Pseudocode describing the insertion approach. The parameters are a single sentence, a desired POS tag, some value of k, and finally min and max number of `[MASK]` tokens inserted. The approach is tailored for high-resource languages, such as English, and uses additional linguistic resources (here, a part of speech tagger) to conduct further filtering.

function HIGHRESOURCEINSERTION($Sentence$, PoS, k, $minMasked$, $maxMasked$)
  $WordsAndScores \leftarrow \emptyset$
  for $n \in [minMasked, maxMasked]$ **do**
    $MaskedSentence \leftarrow$ Sentence with n `[MASK]` tokens inserted
    $Words, Scores \leftarrow$ MASKEDLM.TOPKPREDICTIONS($MaskedSentence, k$)
    $WordsAndScores \leftarrow WordAndScores \cup \{(w, s) | w \in Words$ **and** $s \in Scores\}$
  end for
  return SAMPLE($\{w | (w, s) \in WordsAndScores$, POSTAG$(w) = PoS, s >= Threshold\}$)
end function

Step 1: In Austria in 2018 75 year old or older females {*empty, pos=RB*} received median equivalised net income of 22234 €.

Step 2: In Austria in 2018 75 year old or older females *still* received median equivalised net income of 22234 €.

Figure 1: The general idea of sentence modification using the insertion method. Step 1 represents the intermediate step between a template and the final modified sentence presented in Step 2.

example of the general idea is shown in Figure 1.

During variety induction, a contextual language model with a masked language modeling head (In this case, FinEstBert by Ulčar and Robnik-Šikonja, 2020) is used to predict suitable content to replace the placeholder token. This is achieved by replacing the placeholder token with one or more `[MASK]` tokens in the sentence. Multiple `[MASK]` tokens are required where the language model uses subword tokens. The language model is then queried for the k most likely (subword) token sequences to replace the sequence of `[MASK]` tokens. This results in a selection of potential tokens ('proposals', each consisting of one or more subword tokens) to replace the original placeholder.

As an additional method for control, we associate the original placeholder token with a certain POS tag, and filter the generated proposals to those matching this POS tag. In addition, we use a threshold likelihood value so that each proposal has to reach a minimal language model score. This is required for cases wherein a certain length sequence of mask tokens results in no believable proposals in the top-k selection. Finally, we sample one of the filtered proposals and replace the original placeholder token with it. In cases where there are no suitable proposals, the placeholder value is simply removed. This method is described in pseudocode in Algorithm 1.

Naturally, this approach is dependent on the availability of two linguistic resources: the contextual word embeddings and a POS tagging model. While word embeddings/language models are relatively easily trainable as long as there are any available text corpora, high-quality POS tagging models are less common outside of the most widely spoken languages. To extend this approach to such low-resource languages that have available corpora for training language models such as BERT, but lack POS tagging models, cross-lingual aligned word embeddings can be utilized.

Once a low-resource language proposal has been obtained using the method described above, an aligned cross-lingual word embeddings model – in our case, FastText embeddings (Bojanowski et al., 2016) aligned using VecMap (Artetxe et al., 2018) – between the low-resource language and some high-resource language (e.g. English) can be used to obtain the closest high-resource language token in the aligned embedding space. The retrieved high-reasource language token is, in theory, the closest semantic high-resource language equivalent to the low-resource token. We then apply a POS tagging model *for the high-resource language* to the high-resource 'translation', and use that POS tag as the low-resource token's POS tag for the purposes of filtering the proposals. This approach is described as pseudocode in Algorithm 2.

Algorithm 2 Pseudocode describing how the language resources, here a POS tagger, are utilized for a low-resource language with cross-lingual word embeddings. In other words, when working with a low-resource language, insertion is done as in Algorithm 1, but the POS tagging phase utilises this algorithm. The FINDVECTOR method finds the word embedding vector for the low resource word, and the CLOSESTWORD method is then used for finding the closest match for that vector from the aligned high-resource language embedding space. The algorithm parameters are the low-resource original word to be replaced, and the pairwise aligned low- and high-resource word embeddings.

function POSTAGLOWRESOURCELANGUAGE($LowResWord$, $LowResEmbeddings$, $HighResEmbeddings$)
  $LowResVector \leftarrow$ FINDVECTOR($LowResWord, LowResEmbeddings$)
  $HighResWord \leftarrow$ CLOSESTWORD($LowResVector, HighResEmbeddings$)
  $LowResTagged \leftarrow (LowResWord,$ POSTAG($HighResWord$))
  return $LowResTagged$
end function

Step 1: In Finland in 2016 households' total {*expenditure, replace=True*} on healthcare was 20.35 %.

Step 2: In Finland in 2016 households' total *spending* on healthcare was 20.35 %.

Figure 2: The general idea of sentence modification using the replacement method. Step 1 represents the intermediate step between a template and the final modified sentence presented in Step 2.

3.2 Inducing Variety with Replacement

In addition to insertion of completely new words, variety can also be induced by replacing existing content, so that previously lexicalized words within the text are replaced by suitable alternatives. We propose to use a combination of a synonym dictionary and a contextual language model to do this in a controlled fashion. A simplified example of this approach is shown in Figure 2.

On a high level, we mark certain words within the template fragments used by our system as potential candidates for replacement. This provides us with further control, allowing us to limit the variety induction to relatively 'safe' words such as those not referring to values in the underlying data.

During variation induction, the synonym dictionary is first queried for synonyms of the marked word. To account for homonymy, polynymy, as well as the contextual fit of the proposed synonyms, we then use the contextual word embeddings (with a masked language model head) to score the proposed words. To score the word, it needs to be tokenized. In cases where the word is not part of BERT's fixed size vocabulary, it is tokenized as multiple subword tokens. To account for this we use the mean score of the (subword) tokens as the score of the complete word.

As above, a threshold is used to ensure that only candidates that are sufficiently good fits are retained in the pool of proposed replacements. The final word is sampled from the filtered pool of proposals. If the pool of proposed words is empty after filtering, the sentence is not modified. The original word is also explicitly retained in the proposals. This procedure is shown in Algorithm 3.

We emphasize that the use of the synonym dictionary is required to avoid predicting antonyms, as both antonyms and synonyms reside close to the original word in the word embedding space. While an antonym such as 'increase' for the verb 'decrease' would be a good replacement in terms of language modeling score, such antonymous replacement would change the sentence meaning tremendously and must be prevented.

The modification of the replacement approach for low-resource languages (where no synonym dictionary is available) is similar to that presented above for insertion: We conduct a round-trip via a high-resource language using the cross-lingual embeddings when retrieving synonyms. The low-resource language words are 'translated' to the high-resource language using the cross-lingual embeddings, after which synonyms for these translations are retrieved from the synonym dictionary available in the high-resource language. The synonyms are then 'translated' back to the low-resource language using the same cross-lingual embeddings. This approach is shown in Algorithm 4.

Algorithm 3 Pseudocode describing a method for replacement using a combination of a masked language model (based on contextual word embeddings) and a synonym dictionary, such as provided by WordNet. The parameters are the original word marked to be replaced in the input sentence ('expenditure' in Figure 2), and the input sentence for context.

function HIGHRESOURCEREPLACEMENT($OriginalWord$, $Sentence$)
  $WordsAndScores \leftarrow \emptyset$
  $Synonyms \leftarrow$ GETSYNONYMS($OriginalWord$)
  for $w \in Synonyms$ **do**
    $CandidateSentence \leftarrow$ Sentence with w replacing the original word
    $CandidateScore \leftarrow$ MASKEDLM.SCORE($CandidateSentence, w$)
    $WordsAndScores \leftarrow WordsAndScores \cup (w, CandidateScore)$
  end for
  return SAMPLE($\{w|(w,s) \in WordsAndScores, s >= Threshold\}$)
end function

Algorithm 4 Pseudocode describing how synonyms are retrieved for a low-resource language by utilizing cross-lingual word embeddings. Low-resource variant of replacement is as Algorithm 3, but this algorithm is used to retrieve synonyms. The FINDVECTOR method finds the correct word embedding vector for the low resource word, and the CLOSESTWORD method is then used for finding the closest match for that vector from the aligned high-resource language embedding space. The algorithm parameters are the low-resource original word to be replaced, and the pairwise aligned low- and high-resource word embeddings.

function SYNONYMSFORLOWRESOURCELANGUAGE($LowResWord$, $LowResEmbeddings$, $HighResEmbeddings$)
  $LowResVector \leftarrow$ FINDVECTOR($LowResWord, LowResEmbeddings$)
  $HighResWord \leftarrow$ CLOSESTWORD($LowResVector, HighResEmbeddings$)
  $HighResSynonyms \leftarrow$ GETSYNONYMS($HighResWord$)
  $LowResSynonyms \leftarrow \emptyset$
  for $w \in HighResSynonyms$ **do**
    $HighResVector \leftarrow$ FINDVECTOR($w, HighResEmbeddings$)
    $LowResWord \leftarrow$ CLOSESTWORD($HighResVector, LowResEmbeddings$)
    $LowResSynonyms \leftarrow LowResSynonyms \cup \{LowResWord\}$
  end for
  return $LowResSynonyms$
end function

As we conduct our case study using Finnish as the (simulated) low-resource language, words need to be lemmatized before synonym lookup. We apply UralicNLP (Hämäläinen, 2019) to analyze and lemmatize the original word and reinflect the retrieved synonyms after lookup. A difficulty is presented by the fact that oftentimes, a specific token can have multiple plausible grammatical analyses and lemmas. In our approach, synonyms are retrieved for all of the plausible lemmas, and the algorithm regenerates all morphologies proposed by UralicNLP for all synonyms. While this results in some ungrammatical or contextually incorrect tokens, we rely on the language model to score these as unlikely.

4 Evaluation

We have implemented the above algorithms within a multi-lingual (Finnish and English) natural language generation system that conducts automated journalism from time-series data provided by Eurostat (the statistical office of the European Union). The system is derived from the template-based modular architecture presented by Leppänen et al. (2017). It produces text describing the most salient factors of the input data in several languages in a technically accurate manner using only a few templates, but the resulting language is very stiff, and the sentences are very alike. This makes the final report very repetitive and thus a good candidate for

variety induction.

For all of the algorithms described, we utilise the same trilingual BERT model: FinEst BERT (Ulčar and Robnik-Šikonja, 2020). The FinEst BERT model is trained with monolingual corpora for English, Finnish and Estonian from a mixture of news articles and a general web crawl. In addition to the BERT model, the low-resource language variants of the algorithms utilize cross-lingual pairwise aligned word embeddings for word 'translations'. We use monolingual FastText (Bojanowski et al., 2016) word embeddings mapped with VecMap (Artetxe et al., 2018) to form the cross-lingual embeddings. POS tagging is done with NLTK (Bird et al., 2009) and the lexical database used as a synonym dictionary is WordNet (Miller, 1995).

A human evaluation of our methods was conducted following the best practices proposed by van der Lee et al. (2019). In the evaluation setting, judges were first presented with three statements about a sentence pair. Sentence 1 of the pair was an original sentence, generated by the NLG system without variation induction. Sentence 2 of the pair was the same sentence with a variation induction procedure applied. Cases where the sentence would remain unchanged, or where no insertion/replacement candidates were identified, were ruled out from the evaluation set. The part of the sentence to be modified was marked in the original sentence and the inserted/replaced word highlighted.

The judges were asked to evaluate the following statements on a Likert scale ranging from 1 ('Strongly Disagree') to 4 ('Neither Agree nor Disagree') to 7 ('Strongly Agree'):

Q1: Sentence 1 is a good quality sentence in the target language.

Q2: Sentence 2 is a good quality sentence in the target language.

Q3: Sentences 1 and 2 have essentially the same meaning.

In addition to the two sentences, the judges were presented with two groups of words to examine if using the scores by BERT would correctly distinguish suitable words from unsuitable words. Group 1 contained the words scored as acceptable by BERT while group 2 contained the words ruled out due to a low score. All words in both groups met the criteria of being synonyms (in the case of replacement) or being the correct POS (in the case of insertion). The judges were asked to evaluate the following questions on a 5-point Likert scale ranging from 1 ('None of the words') to 3 ('Half of the words') to 5 ('All of the words'):

Q4: How many of the words in word group 1 could be used in the marked place in sentence 1 so that the meaning remains essentially the same?

Q5: How many of the words in word group 2 could be used in the marked place in sentence 1 so that the meaning remains essentially the same?

For the high-resource language results, we gathered 3 judgements each for 100 sentence pairs. The judges were recruited from an online crowdsourcing platform and they received a monetary reward for participating in the study. The judge recruitment was restricted to countries where majority of people are native speakers of English. For the low-resource language results, 21 judges evaluated 20 sentence pairs. The judges were recruited via student mailing lists of University of Helsinki in Finland and were not compensated monetarily. All but one of the participants in the low-resource evaluation were native speakers of the target language. The final participant self-identified as having a 'working proficiency.'

5 Results

Table 1 presents our results in applying both the insertion and replacement methods to both a high-resource language (English) and a low-resource language (Finnish).

In the high-resource insertion case, the results indicate that inducing variation using the proposed method does not decrease output quality, as both the original sentences' qualities (Q1 mean 5.57) and modified sentences' qualities (Q2 mean 5.76) were similar. As the sentence meaning also remained largely unchanged (Q3 mean 5.54), we interpret this result as a success. The results for Q4 and Q5 indicate that our filtering method based on a threshold language model score can be improved: results for Q4 (mean 3.11 on a 5-point Likert scale) indicate that unsuitable words are left unfiltered, while Q5 (mean 3.03) indicates that some acceptable words are filtered out.

			Insertion		Replacement	
	Range	Statement	En	Fi	En	Fi
Q1	(1–7 ↑)	'Sentence 1 is a good quality sentence in the target language'	5.57 (1.46)	6.43 (0.88)	5.55 (1.46)	6.67 (0.66)
Q2	(1–7 ↑)	'Sentence 2 is a good quality sentence in the target language'	5.76 (1.41)	5.12 (1.36)	5.60 (1.40)	3.89 (1.43)
Q3	(1–7 ↑)	'Sentences 1 and 2 have essentially the same meaning'	5.54 (1.36)	4.34 (1.61)	5.65 (1.27)	3.39 (1.30)
Q4	(1–5 ↑)	'How many of the words in word group 1 could be used in the marked place in sentence 1 so that the meaning remains essentially the same?'	3.11 (1.49)	2.53 (0.82)	3.39 (1.31)	1.76 (0.78)
Q5	(1–5 ↓)	'How many of the words in word group 2 could be used in the marked place in sentence 1 so that the meaning remains essentially the same?'	3.03 (1.41)	1.46 (0.62)	3.21 (1.27)	1.62 (0.76)

Table 1: Evaluation results for the insertion and replacement approaches. English ('En') examples were generated using the high-resource variations, while the Finnish ('Fi') examples were generated using the low-resource variations. Arrows in the range column indicate whether higher (↑) or lower (↓) values indicate better performance. Values are the mean evaluation result and the standard deviation (in parentheses). In the context of the statements, sentence 1 is the original, unmodified sentence, while sentence 2 is a sentence with added variety.

In the low-resource case insertion, we observe some change in meaning (Q3 mean value 4.34) and a slight loss of quality, but even after variety induction the output quality is acceptable (Q1 mean 6.43 vs. Q2 mean 5.12). Interestingly, in the low-resource setting, we observe that the language model is slightly better at distinguishing between suitable and unsuitable candidates (Q4 and Q5 means 2.53 and 1.46, respectively) than in the high-resource case. We are, at this point, uncertain of the reason behind the difference in the ratios of Q4 and Q5 answers between the high-resource and the low-resource case. Notably, even this 'better' result is far from perfect.

We also conducted POS tag specific analyses for both the high-resource and the low-resource insertion cases. In the high-resource case, no major differences were observed between various POS tags. In the low-resource (Finnish) case, however, we observed that with some POS tags, such as adverbs, the results are similar to those observed with English. Low-resource results for adverbs only are shown in Figure 3. We emphasize that this is the best observed subresult and should be viewed as post-hoc analysis.

In the high-resource replacement case, we observe promising results. Inducing variation did not negatively affect sentence quality (Q1 mean 5.55 vs. Q2 mean 5.60) and concurrently retained meaning (Q3 mean 5.65). Results for Q4 and Q5 (means 3.39 and 3.21, respectively) indicate that, as above, the filtering method still has room for improvement, with poor quality options passing the filter and high-quality options being filtered out.

However, in the low-resource case replacement case, we observe a significant drop in sentence quality after variation induction (Q1 mean 6.67 vs Q2 mean 3.89), as well as significant change in sentence meaning (Q3 mean 3.39). While Q5 results are relatively good (mean 1.62), as in very few if any good candidate words are filtered out, Q4 results (mean 1.76) indicate some fundamental problem in the candidate generation process: as there are few if any good candidates in either group, it seems that most of the proposed words are unsuitable.

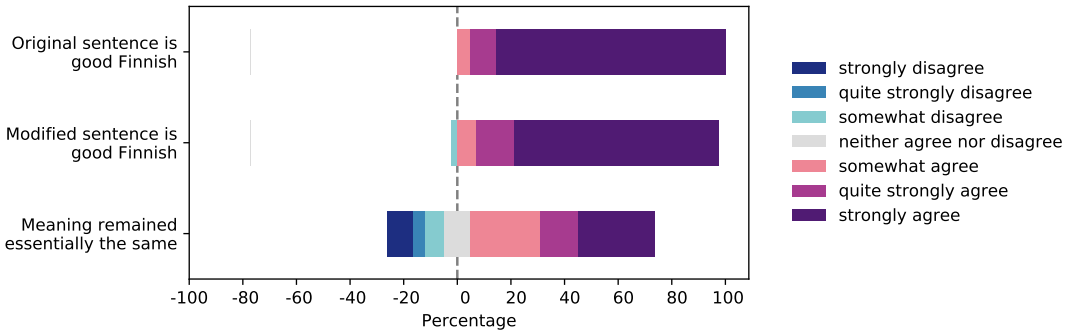

Figure 3: Quality of sentences with low-resource insertion in Finnish with English as the high-resource language, and preservation of sentence meaning. Results shown for adverbs only, representing the best observed performance across the various parts of speech generated. We emphasize that the graph shows only a subset of the complete results (See Table 1), identified as best-performing during post-hoc analysis.

6 Discussion

Our high-resource results indicate that the proposed approach is suitable for inducing some light variation into automatically generated language. The use of synonym dictionaries removes the need to manually construct variants into the templates used in the generation, while the use of language models allows for contextual scoring of the proposed variants so that higher quality results are selected.

We suspect that a major contributor to the low quality of the modified sentences in the low-resource scenarios was the complex morphology of the Finnish language. Especially in the case of Finnish, the process wherein the original word was grammatically analyzed and the replacement word reinflected into the same form would have likely resulted in cases where the resulting word is technically grammatically feasible in isolation, but not grammatical in the context of the rest of the sentence. Our post-hoc investigation also indicates that at least in some cases the resulting reinflected words were outright ungrammatical.

In addition, it seems that the language model employed did not successfully distinguish these failure cases from plausible cases, which led to significant amounts of ungrammatical words populating the proposed set of replacement words. Our post-hoc analysis further indicates that the methods led to better results when use of compound words was avoided in the Finnish templates. We hypothesize that applying the method to a morphologically less complex language might yield significantly better results.

At the same time, in the case of low-resource variation induction using insertion, our results indicate that some success could be found if the method is applied while restrained to certain pre-screened parts of speech, such as adverbs (See Figure 3). This further indicates that the performance of the replacement approach might be improved significantly if the morphology issues were corrected.

Notably, our analysis of the results did not include an in-depth error analysis to determine what parts of the relatively complex procedure fundamentally caused the errors, i.e. were the errors introduced during POS-tagging, language model based scoring, or some other stage. Furthermore, we did not rigorously analyze whether the generation errors were semantic or grammatical in nature.

As a final note, we emphasise that these results were evaluated on local (sentence) rather than on global (full news report) level. We anticipate that, for example, when inserting a word like ‘still’ in a sentence (see Figure 1), the results might differ when evaluating on a global level.

7 Conclusions

In this work, we proposed two approaches, with variations for both high-resource and low-resource languages, for increasing the variety of language in NLG system output in context of news, and presented empirical results obtained by human evaluation. The evaluation suggests that the high-resource variants of our approaches are promising: using them in the context of a case study did create variety, while preserving quality and meaning. The low-resource variants did not perform as well, but we show that there are some positive glimpses in these initial results, and suggest future improvements.

Acknowledgements

This article is based on the Master's thesis of the first author. The work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). We thank Matej Ulčar and Marko Robnik-Šikonja for the VecMap alignment of the FastText embeddings.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR*, abs/1607.04606.

Matt Carlson. 2015. The robotic reporter: Automated journalism and the redefinition of labor, compositional forms, and journalistic authority. *Digital journalism*, 3(3):416–431.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135.

Alexander Fanta. 2017. Putting Europe's robots on the map: Automated journalism in news agencies. *Reuters Institute Fellowship Paper*, 9.

Albert Gatt and Emiel Krahmer. 2017. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61.

Eli Goldberg, Norbert Driedger, and Richard I Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Andreas Graefe. 2016. Guide to automated journalism.

Mika Hämäläinen. 2019. UralicNLP: An NLP library for Uralic Languages. *Journal of Open Source Software*, 4(37):1345.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-Driven News Generation for Automated Journalism. In *The 10th International Natural Language Generation conference, Proceedings of the Conference*, pages 188–197, United States. The Association for Computational Linguistics.

Carl-Gustav Linden. 2017. Decades of automation in the newsroom: Why are there still so many jobs in journalism? *Digital journalism*, 5(2):123–140.

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

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459.

Vassilis Plachouras, Charese Smiley, Hiroko Bretz, Ola Taylor, Jochen L Leidner, Dezhao Song, and Frank Schilder. 2016. Interacting with financial data using natural language. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1121–1124.

Ehud Reiter. 2019. ML is used more if it does not limit control. https://ehudreiter.com/2019/08/15/ml-limits-control/. Accessed: 2020-07-25.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Stefanie Sirén-Heikel, Leo Leppänen, Carl-Gustav Lindén, and Asta Bäck. 2019. Unboxing news automation: Exploring imagined affordances of automation in news journalism. *Nordic Journal of Media Studies*, 1(1):47–66.

Matej Ulčar and Marko Robnik-Šikonja. 2020. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. *arXiv preprint arXiv:2006.07890*.