Smooth inverse frequency based text data selection for medical dictation

Domonkos Bálint[1,2], Péter Mihajlik[1,3]

[1]Budapest University of Technology and Economics,
Department of Telecommunications and Media Informatics, 1111, Budapest,
Műegyetem rkp. 3.
balintdomonkosjozsef@edu.bme.hu, mihajlik@tmit.bme.hu
[2]SpeechTex Kft., 1181, Madách Imre u. 47.
[3]THINKTech Nonprofit Kft., 2600, Vác, Váczy Pál u. 15.

Abstract. Under-resourced domain problem is significant in automatic speech recognition, especially in small languages such as Hungarian or in fields where data is often confidential such as finance and medicine. We introduce a method using word embedding and smooth inverse frequency (SIF) based distance measurement to filter public domain web corpora. The selection for (medical) domain matching documents can be scaled. The resulted text is used to train an augmented language model for a medical dictation system. We show that using the appropriately scaled selection leads to optimal performance of the ASR system over the baselines where no data augmentation was applied or all the augmentation data was added.

Keywords: data selection, data acquisition, smooth inverse frequency, automatic speech recognition, sentence embedding

1 Introduction

In automatic speech recognition (ASR) - as well as in every machine learning field - we need data that fits well to the later area of usage. In some cases (eg. in healthcare, financial) such data are very difficult and costly to collect, and even if they are available, their amount is far below what is required.

In a typical ASR system there are two main models that require data to train them, the acoustic and the language model. The former is trainable with non-domain-specific (general) data, however the latter cannot produce output words that are not included in the training set, so domain-specific data is essential. Overall this phenomenon results in lower accuracy speech recognition systems, with narrow usability.

To tackle this problem, the usual approach is to add general data to the training set that is not related to the targeted topic. However, this method typically only slightly improves the accuracy of speech recognition, while also can multiply the size of the models and thus the resource requirements.

This paper explores whether general data can effectively be filtered using machine learning methods and domain-specific training data, and whether the filtered data can be used as training data to increase accuracy of an ASR system. The rest of the paper is organised as follows. In Section 2 we wrote about the different data acquisition techniques used during ASR development. Section 3 contains information about the data sets employed here. Section 4 continues with the settings of the experiments carried out in this paper, including the different word embedding models explored and the word embedding aggregation method used. The different settings were evaluated in an independent test set, the results are presented in Section 5. The paper finishes with conclusions in Section 6.

2 Related Works

We can distinguish between three main approaches, addressing this under-resourced domain problem, such as text generation and augmentation, text translation and data crawling. We will briefly introduce them in that order.

Text generation using recurrent neural network (RNN) architectures is a common application that can be used to address this problem (Barzilay and Lapata, 2005), (Koncel-Kedziorski et al., 2019). Creating text of similar nature to the available limited amount of domain specific data is one straightforward way to increase the size of the corpus. Transformer networks are also used for this task (Tarján et al., 2020), as they tend to provide state-of-the-art results in the field of natural language processing (NLP) and especially in text generation (Brown et al., 2020), (Devlin et al., 2018).

One different approach of data augmentation was shown in (Wei and Zou, 2019). Through simple operations like synonym replacement, random insertion, random swap, and random deletion this was able to improve the performance of the examined neural networks on five different NLP tasks.

In the case of end-to-end ASR (E2E) it has been shown that a back-translation (Sennrich et al., 2015), (Lample et al., 2017) style data augmentation could improve its performance (Hayashi et al., 2018). An E2E ASR system is typically an encoder-decoder neural network, and it is only trainable with voice, text pairs (Cho et al., 2014), (Sutskever et al., 2014). This method can utilize unpaired text corpora, as they are used to train only the decoder network.

Language models trained on a machine translated text have been found to be useful in a low-resourced setting (Jensson et al., 2008). Cross-lingual language model pretraining could also effectively be used for under-resourced languages (Lample and Conneau, 2019). Overall we found that medical data is confidential in every developed country, therefore it is similarly difficult to acquire clean data in e.g. English as in Hungarian.

Many publications refer to online text data acquisition as a possible solution (Sethy et al., 2006), (Remus and Biemann, 2016a). One approach is focused web-crawling (Chakrabarti et al., 2000). This term refers to the process of crawling the web in a guided way with focus on a specific topic. Without any labeled data, one proven method is language model and perplexity based crawling (Remus and

Biemann, 2016b). In (Vogel et al., 2020) they used different document similarity measuring methods to automatically collect in-domain texts from the web.

Our work differs from existing approaches as we used traditional crawling to acquire data, and filtered this data to get the final training set. In this way we could find the closest documents to our domain, therefore we did not have to set a hard limit to determine which documents were close enough to our reference. In addition, to the best of our knowledge this kind of document similarity based database creation is not yet published for Hungarian language.

3 Data Sources

The experiments presented in this paper were carried out using several datasets which will be explained in the next subsections.

WebBeteg: To evaluate the accuracy of our models, we needed a database that contained topic-specific data but also many other irrelevant data. For this, we selected the medical question-and-answer section of the WebBeteg Hungarian healthcare site[1].

The resulting database contains nearly 150 000 questions, of which 10% contain medical records (based on manual sampling and evaluation).

From this database, we manually selected 50 documents that contained medical records. We have considered this as the reference set. Our goal was to find similar entries in the rest of the database. This set contains 3256 tokens, its vocabulary size is 1751. As we used our reference set as a test set, we created audio files from the selected texts. The text audio is 36 minutes long.

Proprietary data: A large, but to our specific task only loosely related medical database was already available. This includes medical journals, medication descriptions, and a small amount of X-ray records. The database consists of 1 717 851 unique tokens, overall its size is 60 362 655 tokens.

4 Methods

4.1 Word Embeddings

In order to use words in machine learning models, they have to be represented with a numerical form. Over the years researchers have used many word representations like bag-of-words, one-hot encoded vectors etc. However the recent neural models like word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) provide better representations to the words considering its context, too. Their main weakness is that every word has a unique word embedding regardless of the context it appears. As an example the word 'bank' in two sentences - "I

[1] https://www.webbeteg.hu/orvos-valaszol

am walking by the river bank" and "I deposited money to the bank" would have the same embeddings which can be confusing for machine learning models. The recent introduction of contextualised word representations solved this problem by providing vectors for words considering their context too. In this way the word 'bank' in the above sentences has two different embeddings. As a result, contextualised word embeddings perform better than standard word embeddings in many natural language processing tasks like question answering, textual entailment etc. (Devlin et al., 2018). The following words representation models were considered for the experiments.

FastText FastText is a library for learning word embeddings based on (Bojanowski et al., 2017). This provides a model, which is based on the skip-gram model, where each word is represented as a bag of character n-grams. A vector representation is associated with each character n-gram; words being represented as the sum of these representations. This representation has a very useful effect on a corpus with many rare words (eg. corpora written in an agglutinative language, or a corpus with a lot of misspelled words): similar words have similar representations, if they are in the same context. We used a pretrained model provided in the official fasttext website.[2] It is a 300 dimension embedding, and it was trained on the Hungarian Wikipedia.

ELMo ELMo introduced by (Peters et al., 2018) uses bidirectional language model (biLM) to learn both word (e.g., syntax and semantics) and linguistic context. After pretraining, an internal state of vectors can be transferred to downstream natural language processing tasks. We used a pre-trained Hungarian model provided in (Che et al., 2018), (Fares et al., 2017) which trained on a 20 million sample of WikiDump[3] and Common Crawl[4]. Using the model we represented each word as a vector with a size of 4096 values.

4.2 Smooth Inverse Frequency

We acquired sentence embeddings using Smooth Inverse Frequency (SIF) proposed by (Arora et al., 2017) and then calculated the cosine similarity between those embeddings.

Semantically speaking, taking the average of the word embeddings in a sentence tends to give too much weight to words that are quite irrelevant. Smooth Inverse Frequency tries to solve this problem in two steps.

- Weighting: Smooth Inverse Frequency takes the weighted average of the word embeddings in the sentence:

$$v'_s = \frac{a}{|s|} \sum_{w \in s} \frac{a}{a + P(w)} \cdot v_w \tag{1}$$

[2] https://fasttext.cc/docs/en/pretrained-vectors.html

[3] https://dumps.wikimedia.org/

[4] https://commoncrawl.org/

where s is the input sentence, w is a word in s, v_w is the word embedding of w, $P(w)$ is the estimated frequency of w and a is a parameter that is typically set to 0.001.

- Common component removal: We assume that there is n sentences in the corpus, in the next step SIF creates a matrix from all the previously calculated sentence embeddings:

  $$X = [v'_{s1}|v'_{s2}|...|v'_{sn}] \tag{2}$$

  Then the algorithm computes the principal component of X. It then subtracts their projections on first principal component from these sentence embeddings:

  $$v_s = v'_s - uu^T v'_s \tag{3}$$

  where u is the principal component of X. This should remove variation related to frequency and syntax that is less relevant semantically. v_{s1} is the final sentence embedding output for sentence $s1$.

As a result, Smooth Inverse Frequency downgrades unimportant words such as but, just, etc., and keeps the information that contributes most to the semantics of the sentence. After acquiring the sentence embeddings for a pair of sentences, the cosine similarity between those two vectors were taken to represent the similarity between them.

In (Ranasinghe et al., 2019) they showed that SIF with the previously presented word embeddings can perform in the same level as the much more complex transformer networks on the task of document similarity, therefore we chose these methods in our paper.

4.3 Language modeling

N-gram models Back-off, n-gram language models (BNLMs) are still commonly used in online, single-pass speech transcription systems due to their lower source demand and high compatibility with Weighted Finite-State Transducer (WFST) decoders. Hence we applied BNLMs as our language models in our experiments. All BNLMs are trained with modified Kneser-Ney discounting (Chen and Goodman, 1996) applying the implementation of SRI language modeling toolkit (Stolcke, 2002). We carried out a preliminary experiment on the development set and found 3-gram the optimal LM order for word BNLMs.

5 Results

This section describes the evaluation results of WebBeteg data for all methods we mentioned above. All experiments were evaluated using WER (word error rate), and LER (letter error rate) of the ASR system. We also calculated the perplexity and the out-of-vocabulary (OOV) word count for every language model. The latter gives us a more precise image of the goodness of our text filtering method, as it does not include the noise from the acoustic model. However during our evaluation we considered WER and LER more important metrics, as our final goal was to improve the ASR system and these gave us information about the accuracy of the whole ASR system, not just the language model.

A new language model was trained for every setup, but the acoustic model and the decoding remained the same. The latter two are described thoroughly in (Varga et al., 2015). The following subsections will discuss the results in detail in each case.

5.1 FastText and SIF

We assigned a vector to the texts in the WebBeteg database and to the reference text using the SIF algorithm (Arora et al., 2017) and FastText word embedding (Bojanowski et al., 2017). We then calculated the cosine similarity between the text vectors and the reference vector, and established an order based on these distances. Then, we selected the first n pieces from the database based on the order. The names of the models were created from this n number: for example, in the FastText f100 model, we selected the first 100 items from the WebBeteg database. Table 1 shows the size of the resulting text sets. We created a language model from it, and interpolated with the language model from the proprietary database (see Section 3). The interpolation weights of the two models were 0.5, 0.5, based on a one variable optimization process.

dataset	Token count	Unique token count
Proprietary data	60 362 655	1 717 851
WebBeteg	13 068 749	945 280
WebBeteg f100	20 310	6 802
WebBeteg f1 000	158 851	33 171
WebBeteg f10 000	1 228 734	181 089
WebBeteg f100 000	9 138 986	700 208

Table 1. The token count, and the unique token count of the different databases. (e.g. WebBeteg f100 is the first 100 best text from WebBeteg database - see section 5.1)

Table 2 shows the results for models using the data sorted by FastText and SIF methods as we described above.

5.2 ELMo and SIF

We followed the same algorithm as described in Section 5.1, with the difference that ELMo embedding (Peters et al., 2018) was used instead of FastText embedding.

The naming conventions are the same as before: for example, in ELMo f100 we selected the first 100 elements of the WebBeteg database to train the model. The results are shown in the Table 2.

model name	size	WER (%)	LER (%)	perplexity	OOV (%)
baseline1		27.14	8.36	580	3.34
baseline2		23.34	6.92	406	3.14
FastText	f100	24.03	7.08	398	3.34
	f1 000	**22.86**	**6.57**	295	3.34
	f10 000	23.14	6.98	**278**	3.21
	f100 000	23.21	6.95	379	3.14
ELMo	f100	24.28	7.15	402	3.34
	f1 000	23.28	**6.78**	279	3.31
	f10 000	**22.97**	6.79	**268**	3.24
	f100 000	23.34	6.92	392	3.14

Table 2. The WER, LER, perplexity and OOV results of the different models described in Section 5. Baseline1 and baseline2 are when we add nothing or everything from WebBeteg respectively.

6 Conclusion

In this paper, we filtered an online database to create a smaller in-domain set. We examined whether the WER and LER values of a speech recognizer can be improved by interpolating the language model created from this textset with another language model trained on an orders of magnitude larger database, thereby adapting the resulting language model to the task.

The results (Table 2) show that the recognition metrics of the models after one point begin to decrease with the additional data. This is advantageous as we can improve the model without a huge increase in complexity or size. We also established that a more complex word embedding like ELMo (Peters et al., 2018) can't improve the aforementioned filtering. The reason why there is no significant difference between results of the different embedding techniques applied requires further investigations. Overall we found it useful to perform text similarity based filtering for noisy databases used in speech recognition training.

Bibliography

Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: ICLR (2017)

Barzilay, R., Lapata, M.: Collective content selection for concept-to-text generation. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. pp. 331–338. Association for Computational Linguistics, Vancouver, British Columbia, Canada (Oct 2005), https://www.aclweb.org/anthology/H05-1042

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020)

Chakrabarti, S., Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific web resource discovery. Computer Networks 31, 1623–1640 (04 2000)

Che, W., Liu, Y., Wang, Y., Zheng, B., Liu, T.: Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 55–64. Association for Computational Linguistics, Brussels, Belgium (October 2018), http://www.aclweb.org/anthology/K18-2005

Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics. p. 310–318. ACL '96, Association for Computational Linguistics, USA (1996), https://doi.org/10.3115/981863.981904

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation (2014)

Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018), http://arxiv.org/abs/1810.04805

Fares, M., Kutuzov, A., Oepen, S., Velldal, E.: Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In: Proceedings of the 21st Nordic Conference on Computational Linguistics. pp. 271–276. Association for Computational Linguistics, Gothenburg, Sweden (May 2017), http://www.aclweb.org/anthology/W17-0237

Hayashi, T., Watanabe, S., Zhang, Y., Toda, T., Hori, T., Astudillo, R., Takeda, K.: Back-translation-style data augmentation for end-to-end asr (2018)

Jensson, A., Iwano, K., Furui, S.: Language model adaptation using machine-translated text for resource-deficient languages. EURASIP J. Audio Speech Music Process. 2008(1) (Dec 2008)

Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., Hajishirzi, H.: Text generation from knowledge graphs with graph transformers (2019)

Lample, G., Conneau, A.: Cross-lingual language model pretraining (2019)

Lample, G., Conneau, A., Denoyer, L., Ranzato, M.: Unsupervised machine translation using monolingual corpora only (2017)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR abs/1310.4546 (2013), http://arxiv.org/abs/1310.4546

Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (Oct 2014), https://www.aclweb.org/anthology/D14-1162

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR abs/1802.05365 (2018), http://arxiv.org/abs/1802.05365

Ranasinghe, T., Orasan, C., Mitkov, R.: Enhancing unsupervised sentence similarity methods with deep contextualised word representations. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). pp. 994–1003. INCOMA Ltd., Varna, Bulgaria (Sep 2019), https://www.aclweb.org/anthology/R19-1115

Remus, S., Biemann, C.: Domain-specific corpus expansion with focused webcrawling. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 3607–3611. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016a), https://www.aclweb.org/anthology/L16-1572

Remus, S., Biemann, C.: Domain-specific corpus expansion with focused webcrawling. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 3607–3611. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016b), https://www.aclweb.org/anthology/L16-1572

Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data (2015)

Sethy, A., Georgiou, P.G., Narayanan, S.: Text data acquisition for domain-specific language models. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. pp. 382–389. Association for Computational Linguistics, Sydney, Australia (Jul 2006), https://www.aclweb.org/anthology/W06-1645

Stolcke, A.: Srilm - an extensible language modeling toolkit. In: INTERSPEECH (2002)

Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks (2014)

Tarján, B., Szaszák, G., Fegyó, T., Mihajlik, P.: Deep transformer based data augmentation with subword units for morphologically rich online asr (2020)

Varga, Á., Tarján, B., Tobler, Z., Szaszák, G., Fegyó, T., Bordás, C., Mihajlik, P.: Automatic close captioning for live hungarian television broadcast speech:

A fast and resource-efficient approach. In: Ronzhin, A., Potapova, R., Fakotakis, N. (eds.) Speech and Computer. pp. 105–112. Springer International Publishing, Cham (2015)

Vogel, I., Choi, J.E., Meghana, M.: Similarity detection pipeline for crawling a topic related fake news corpus (2020)

Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks (2019)