Extended Language Modeling Experiments for Kazakh

Bagdat Myrzakhmetov[1,2] and Zhanibek Kozhirbayev[1]

[1] National Laboratory Astana, Nazarbayev University, Astana, 010000, Kazakhstan
[2] Nazarbayev University, School of Science and Technology, Astana, 010000, Kazakhstan
bagdat.myrzakhmetov@nu.edu.kz , zhanibek.kozhirbayev@nu.edu.kz

Abstract. In this article we present dataset for the Kazakh language for the language modeling. It is an analogue of the Penn Treebank dataset for the Kazakh language as we followed all instructions to create it. The main source for our dataset is articles on the web-pages which were primarily written in Kazakh since there are many new articles translated into Kazakh in Kazakhstan. The dataset is publicly available for research purposes[1]. Several experiments were conducted with this dataset. Together with the traditional n-gram models, we created neural network models for the word-based language model (LM). The latter model on the basis of large parameterized long short-term memory (LSTM) shows the best performance. Since the Kazakh language is considered as an agglutinative language and it might have high out-of-vocabulary (OOV) rate on unseen datasets, we also carried on morph-based LM. With regard to experimental results, sub-word based LM is fitted well for Kazakh in both n-gram and neural net models compare to word-based LM.

Keywords: Language Modeling, Kazakh language, n-gram, neural language models, morph-based models.

1 Introduction

The main task of the language model is to determine whether the particular sequence of words is appropriate or not in some context, determining whether the sequence is accepted or discarded. It is used in various areas such as speech recognition, machine translation, handwriting recognition [1], spelling correction [2], augmentative communication [3] and Natural Language Processing tasks (part-of-speech tagging, natural language generation, word similarity, machine translation) [4, 5, 6]. Strict rules may be required depending on the task, in which case language models are created by humans and hand constructed networks are used. However, development of the rule-based approaches is difficult and it even requires costly human efforts if large vocabularies are involved. Also usefulness of this approach is limited: in most cases (especially when a large vocabulary used) rules are inflexible and human mostly produces the ungrammatical sequences of words during the speech. One thing, as [7] states, in most cases the task of language modeling is "to predict how likely the sequence of

[1] https://github.com/Baghdat/LSTM-LM/tree/master/data/

words is", not to reject or accept as in rule-based language modeling. For that reason, statistical probabilistic language models were developed.

A large number of word sequences are required to create the language models. Therefore the language model should be able to assign probabilities not only for small amounts of words, but also for the whole sentence. Nowadays it's possible to create large and readable text corpora consisting of millions of words, and language models can be created by using this corpus.

In this work, we first created the datasets for the language modeling experiments. We built an analogy of the Penn Treebank corpus for the Kazakh language and to do so we followed all preprocessing steps and the corpus sizes. The Penn Treebank (PTB) Corpus [8] is widely used dataset in language modeling tasks in English. The PTB dataset originally contains one million words from the Wall Street Journal, small portion of ATIS-3 material and tagged Brown corpus. Then [9] preprocessed this corpus, divided into training, validation and test sets and restricted the vocabulary size to 10k words. From then, this version of PTB corpus is widely in language modeling experiments for all state of the art language modeling experiments. We made our dataset publicly available for any research purposes. Since there are not so many open source corpora in Kazakh, we hope that this dataset can be useful in the research community.

Various language modeling experiments were performed with our dataset. We first tried traditional n-gram based statistical models, after that performed state-of-the-art Neural Network based language modeling experiments. Neural Network experiments were conducted by using the LSTM [10] cells. LSTM based neural network with large parameters showed the best result. We evaluated our language modeling experiments with the perplexity score, which is a widely used metric to evaluate language models intrinsically. As the Kazakh language is agglutinative language, word based language models might have high portion of out of vocabulary (OOV) words on unseen data. For this reason, we also performed morpheme-based language modeling experiments. Sub-word based language model is fitted well for Kazakh in both n-gram and neural net models compare to word-based language models.

2 Data preparation

We collected the datasets from the websites by using our manual Python scripts, which uses BeautifulSoup and Request libraries in Python. These collected datasets were parsed with our scripts on the basis of the HTML structure. The datasets were crawled from 4 web-pages, whose articles originally written in Kazakh: `egemen.kz`, `zhasalash.kz`, `anatili.kazgazeta.kz` and `baq.kz`. These web-pages mainly contain news articles, historical and literature texts. There are many official web-pages in Kazakhstan which belong to state bodies and other quasi-governmental establishments where texts in Kazakh could be collected. However, in many cases, these web-pages provide the articles, which were translated from the Russian language. In these web-pages, the news articles at the beginning will be written in Russian, only then, these articles translated into Kazakh. These kind of datasets might not well show the inside nature of the Kazakh language, as during the translation, the structure of the sentences and the use of words changes. We barely see the resistant phraseological units of Kazakh in these translated articles, instead we might see the translated version of the phraseological texts in other language. [11] studied original and translated texts in Machine translation, and found out that original texts might be significantly differing from the original texts. For this reason, we excluded the web-pages which might have translation texts. We choose the web-pages whose texts originally written in Kazakh. The statistics of datasets is given in Table 1.

Table 1. Statistics of the dataset: train, validation and test sets shown separately for each source.

Sources	# of documents	# of sentences	# of words
`egemen.kz`	950/80/71	21751/1551/1839	306415/22452/26790
`zhasalash.kz`	1126/83 /95	8663/694/751	102767/8188/9130
`anatili.kazgazeta.kz`	438/32/37	23668/1872/2138	311590/23703/27936
`baq.kz`	752/72/74	13899/1082/1190	168062/13251/14915
Overall	3266/267/277	67981/5199/5918	886872/67567/78742

After collection of the datasets, we preprocessed the datasets by following [9]. First, all collected datasets were tokenized using Moses [12] script. We added non-breaking prefixes for Kazakh in Moses, so as not to split the abbreviations. Next preprocessing steps involved: lowercasing, normalization of punctuations. After normalization of the punctuations, we removed all punctuation signs. All digits were replaced by a special sign “N”. We removed all sentences whose length is shorter than 4 and longer than 80 words and also duplicate sentences. After these operations, we restricted the vocabulary size with 10000: we found the most frequent 10000 words and then replaced all words with ‘<unk>’, which are not in the list of the most frequent words.

After preprocessing of the datasets, we divided our datasets into training, validation and testing sets. We tried to follow the size of the Penn Treebank corpus. Since our datasets were built from the four sources, we tried to split all sources in the same proportion into training, validation and test sets. Since, the contents in each source might differ (for example, in `egemen.kz` there are mostly official news, on the other hand `anatili.kazgazeta.kz` contains mainly historic, literature articles), we avoid having one source as training and others only for testing or validation. For this reason, we split each source with equal portions. Our datasets divided into training, validation and test sets on the document level. The statistics about training, validation and test sets is given in Table 2. Note, overall sentence and word numbers might not be the sum of all columns, because we exclude the repeated sentences. To compare the size, at the end, we provide the statistics of the Penn Treebank corpus.

Table 2. Statistics about the training, validation and test sets.

Sources	Train set	Validation set	Test set
`egemen.kz`	306415	22452	26790
`zhasalash.kz`	102767	8188	9130
`anatili.kazgazeta.kz`	311590	23703	27936
`baq.kz`	168062	13251	14915
Overall	886872	67567	78742
Penn Tree Bank dataset	887521	70390	78669

3 n-gram based models

The main idea behind the language modeling is to predict hypothesized word sequences in the sentence with the probabilistic model. “N-gram models predict the next word from the previous N-1 words” and it is an N-token sequence of words, [13] for example, if we say two-gram model (or more often it is called a bigram model) it is two-word sequence such as “Please do”, “do your”, “your homework” and three gram model consists of the three-word sequences and so on. As [13] states, in n-gram model, the model computes the following word from the preceding. The N-gram idea can be formulated as: given the pervious word sequence and find the probability of the next words. During the computing of probabilities of the word sequences it’s important to define the boundaries (punctuation marks such as period, comma, column or starting of the new sentence from the new line) in order to prevent the search from being computationally unmanageable.

Formulated mathematically, the goal of a language model is to find the probability of word sequences, $P(w_1, ..., w_n)$, and it can be estimated by the chain rule of a probability theory:

$$P(w_1, ..., w_n) = P(w_1)\times P(w_2|w_1)\times\ldots\times P(w_n|w_1, ..., w_{n-1}) \quad (1)$$

There is a notion about history, for example, in the case $P(w_4|w_1, w_2, w_3)$, (w_1, w_2, w_3) considered as the history. This probability is found based on frequency.

We can write the formula for all cases bigram and trigram models as:

$$P(w_i|w_1...w_{i-1}) \approx P(w_i|w_{i-1}) \quad (2)$$

$$P(w_i|w_1...w_{i-1}) \approx P(w_i|w_{i-2}w_{i-1}) \quad (3)$$

This assumption helps to reduce the computation and allows probabilities to be estimated for a large corpus. Also the assumption probability of the word which depends on the previous n words (or previous 3 words for a trigram) is called a **Markov assumption**. This Markov model [14] assumes that it is possible to predict the probability of some future cases without looking deeply into the past.

By using a Markov assumption, we can find the probability of the sequence of words by the following formula:

$$\mathrm{P}(w_1,\ \ldots,\ w_n) = \prod P(w_i|w_1 \ldots w_{i-1}) \approx \prod P(w_i|w_{i-1}) \tag{4}$$

for bigram model and for trigram:

$$\approx \prod (w_i|w_{i-2}w_{i-1}) \tag{5}$$

Up to recently, n-gram language models widely used in all language modeling experiments. In Kazakh, n-gram based language models still used in Speech Processing [15] and Machine translation [16] tasks. We trained n-gram models with the SRILM toolkit [17] with adding 0 smoothing technique. For our dataset, using of the modified Kneser-Ney [18] or Katz backoff [19] algorithms showed poor results, (543.63 on the test set), as there are many infrequent words replaced by '<unk>' sign, and only high gram models might work well. Adding 0 smoothing technique showed best performance for n-gram models. The results are given in Table 3.

4 Neural LSTM based models

In this experiment, we performed Neural LSTM-based language models. There are many types of neural architectures, which also applied successfully for the language modeling tasks. Starting from the work of [20] there are many Recurrent Neural Architectures proposed. With Recurrent Neural Networks, it's possible to model the word sequences, as the recurrence allows to remember the previous word history. Recurrent Neural Network can directly model the original conditional probabilities:

$$\mathrm{P}(w_1,\ \ldots,\ w_n) = \prod P(w_i|w_1 \ldots w_{i-1}) \tag{6}$$

To model the sequences, f function constructed via recursion, initial condition is given by h0 = 0 and the recursion will be ht=f(xt, ht−1). Here, ht is called hidden state or memory and it memorizes the history from x1 up to xt−1. Then, the output function is defined by combination of ht function:

$$\mathrm{P}(w_1,\ \ldots,\ w_n) = g_w(h_t) \tag{7}$$

f can be any nonlinear function such as *tanh*, *ReLU* and *g* can be a *softmax* function.

In our work, we followed [21] who presented a simple regularization technique for Recurrent Neural Networks (RNNs) with LSTM [10] units. [22] proposed dropout technique for regularizing the neural networks, but this technique does not work well with RNNs. This regularizing technique is tent to have overfitting in many tasks. [21] showed that the correctly applied dropout technique to LSTMs might substantially reduce the overfitting in various tasks. They tested their dropout techniques on language modeling, speech recognition, machine translation and image caption generation tasks.

In general, LSTM gates' equations given as follow:

$$f_t = \sigma(W_f[C_{t-1},\ h_{t-1},\ x_t] + b_f]) \tag{8}$$

$$i_t = \sigma(W_i[C_{t-1}, h_{t-1}, x_t]+b_i]) \tag{9}$$

$$o_t = \sigma(W_o[C_t, h_{t-1}, x_t]+b_o]) \tag{20}$$

$$g_t = tanh(W_g[C_t, h_{t-1}, x_t]+b_g]) \tag{31}$$

Then the state values computed by using the above gates:

$$c^l_t = f \odot c^l_{t-1} + i \odot g \tag{42}$$

$$h^l_t = o \odot tanh(c^l_t) \tag{53}$$

The dropout method by [21] can be described as follows: if there is a dropout operator, then it forces the intermediate computation to be more robustly, as the dropout operator corrupts the information carried by the units. On the other hand, in order not to erase all the information from the units, the units remember events that occurred many time steps in the past.

We also implement our[2] LSTM based Neural Network models using TensorFlow [23]. We trained regularized LSTMs of three sizes: the small LSTM, medium LSTM and large LSTM. Small sized model has two layers and unrolled for 20 steps. Medium and large LSTMs have two layers and are unrolled for 35 steps. Hidden size differs in three models: 200, 650 and 1500 for small, medium and large models respectively. We initialize the hidden states to zero. We then use the final hidden states of the current minibatch as the initial hidden state of the subsequent minibatch.

Our experiments showed that the LSTM based neural language modeling outperforms the n-gram based models. Large and Medium LSTM models shows better results than the n-gram add 0 smoothing method (Note, for n-gram Kneser-Ney discounting method we got poor results). Our experiments show that the using of the Neural based language models have better performance for Kazakh. The results are given in Table 3.

Table 3. Word-based language modeling results.

	n-gram	Neural LM		
		small	medium	large
Train ppl	93.81	68.522	67.741	**63.185**
Validation ppl	129.6537	143.871	118.875	**113.944**
Test ppl	123.7189	144.939	118.783	**115.491**

5 Sub-word based language models

In the last section, we experimented with the sub-word based language models. The Kazakh language as other Turkic languages is an agglutinative language, the word forms can be obtained by adding the prefixes. This agglutinative nature may lead on

[2] https://github.com/Baghdat/LSTM-LM

having the high degree of the out-of-vocabulary (OOV) words on unseen data. To solve this problem, depending on the characteristics of individual languages, different language model units were proposed. [24] studied different word representations, such as morphemes, word segmentation based on the Byte Pair Encoding (BPE), characters and character trigrams. Byte Pair Encoding, proposed by [25], can effectively handle rare words in Neural Machine Translation and it iteratively replaces the frequent pairs of characters with a single unused character. Their experiments showed that for fusional languages (Russian, Czech) and for agglutinative languages (Finnish, Turkish) character trigram models perform best. Also, [26] considered syllables as the unit of the language models and tested with different representational models (LSTM, CNN, summation). As they stated, syllable-aware language models fail to outperform character-aware ones, but usage of syllabification can increase the training time and reduce the number of parameters compared to the character-aware language models.

By considering these facts, in this section we experimented with the sub-word based models. Morfessor [27] is a widely tool to split the datasets into morpheme-like units. It used successfully in many agglutinative languages (Finnish, Turkish, Estonian). As for now, there is no syllabification tool for Kazakh, we also used Morfessor tool to split our datasets into morpheme like units.

After splitting the datasets, we performed language modeling experiments on morpheme like units. The results are given in Table 4. By looking at the results, we can say that splitting the words into morpheme-like units benefits in terms of OOV and perplexity in both n-gram and neural net based models.

Table 4. Morph-based language modeling results.

	n-gram	Neural LM		
		small	medium	large
Train ppl	32.39255	19.599	24.999	25.880
Validation ppl	44.11561	50.904	41.896	40.876
Test ppl	44.39559	47.854	38.180	**37.556**

6 Conclusion

In this work we created analogy of the Penn TreeBank corpus for the Kazakh language. To create the corpus, we followed all instructions for preprocessing and the size of the training, validation and test sets. This dataset is publicly available for the research purposes. We conducted language modeling experiments on this dataset by using the traditional n-gram and LSTM based neural networks. We also explored the sub-word units for the language modeling experiments for Kazakh. Our experiments showed that neural based models outperform the n-gram based models and splitting the words into morpheme-like units has advantage compared to the word based models. In future, we are going to create the hyphenation tool for the Kazakh language, as Morfessor's morpheme-like units are data-driven and sometimes there are incorrect morpheme-like units.

Acknowledgement

This work has been funded by the Nazarbayev University under the research grant No129-2017/022-2017 and by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan under the research grant AP05134272.

References

1. Russell S. and Norvig P. Artificial Intelligence: A Modern Approach (2nd Ed.). Pretice Hall. 2002.
2. Kukich K. Techniques for automatically correcting words in text. ACM Computing Surveys.1992. 24(4), pp. 377-439.
3. Newell A., Langer S. and Hickey M. The role of natural language processing in alternative and augmentative communication. Natural Language Engineering. 1998. 4(1). pp. 1-16.
4. Church K.W. A stochastic parts program and noun phrase parser for unrestricted text. In Proceedings of the Second Conference on Applied Natural Language Processing. 1988. pp. 136–143.
5. Brown P.F., Cocke J., DellaPietra S.A., DellaPietra V.J., Jelinek F., Lafferty J.D., Mercer R.L., and Roossin P.S. A statistical approach to machine translation. Computational Linguistics. 1990. 16(2). pp. 79–85.
6. Hull J.J. Combining syntactic knowledge and visual text recognition: A hidden Markov model for part of speech tagging in a word recognition algorithm. In AAAI Symposium: Probabilistic Approaches to Natural Language. 1992. pp. 77–83.
7. Whittaker E. W. D. Statistical Language Modelling for Automatic Speech Recognition of Russian and English. PhD thesis, Cambridge University, Cambridge. 2000.
8. Marcus M.P., Marcinkiewicz M.A. and Santorini B. Building a large annotated corpus of English: The penn Treebank. Computational linguistics. 1993. 19(2). pp. 313–330.
9. Mikolov T., Kombrink S., Burget L., Černocký J. and Khudanpur S. Extensions of recurrent neural network language model. In Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. 2011. pp. 5528-5531. IEEE.
10. Hochreiter S. and Schmidhuber J̈. Long short-term memory. Neural computation. 1997. 9(8). pp. 1735–1780.
11. Lembersky G., Ordan N. and Wintner S. Language models for machine translation: Original vs. translated texts. Computational Linguistics. 2012. 38(4). pp. 799-825.
12. Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R. and Dyer C. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions. Association for Computational Linguistics. 2007. pp. 177-180.
13. Jurafsky D. and Martin J. H. Speech and Language Processing (2nd Ed.). Pretice Hall. 2009.
14. Markov A.A. Primer statisticheskogo issledovaniya nad tekstom “Evgeniya Onegina”, illyustriruyushchij svyaz’ ispytanij v tsep’. [Example of a statistical investigation illustrating the transitions in the chain for the “Evgenii Onegin” text.]. Izvestiya Akademii Nauk. 1913. pp. 153-162.

15. Kozhirbayev Zh, Karabalayeva M. and Yessenbayev Zh. Spoken term detection for Kazakh language. In Proceedings of the 4-th International Conference on Computer Processing of Turkic Languages "TurkLang 2016". 2016. pp. 47-52.
16. Myrzakhmetov B. and Makazhanov A. Initial Experiments on Russian to Kazakh SMT. Research in Computing Science. 2017. vol. 117. pp. 153–160.
17. Stolcke, A. SRILM – an extensible language modeling toolkit. In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP). 2002. pp. 901–904. URL: http://www.speech.sri.com/ projects/srilm/.
18. Kneser R. and Ney H. Improved backing-off for m-gram language modeling. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. 1995. vol. 1. pp. 181-184.
19. Katz S. M. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing. 1987. 35(3). pp. 400-401.
20. Bengio Y., Ducharme R., Vincent P. & Jauvin C. A neural probabilistic language model. Journal of machine learning research. 2003. pp. 1137-1155.
21. Zaremba W., Sutskever I. and Vinyals O. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329. 2014.
22. Srivastava N., Hinton G., Krizhevsky A., Sutskever I. & Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research. 2014. 15(1). pp. 1929-1958.
23. Abadi M., Barham P., Chen J., Chen Zh., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M. and Kudlur M. Tensorflow: a system for large-scale machine learning. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation. USENIX Association. 2016. pp. 265–283.
24. Vania, C., & Lopez, A. From Characters to Words to in Between: Do We Capture Morphology? In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017. Volume 1: Long Papers. Vol. 1, pp. 2016-2027.
25. Sennrich, R., Haddow, B., & Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. Volume 1: Long Papers. Vol. 1, pp. 1715-1725.
26. Assylbekov Z., Takhanov, R., Myrzakhmetov, B., & Washington, J. N. Syllable-aware Neural Language Models: A Failure to Beat Character-aware Ones. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017. pp. 1866-1872.
27. Smit P., Virpioja S., Grönroos S. A. & Kurimo M. Morfessor 2.0: Toolkit for statistical morphological segmentation. In The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April 26-30, 2014. Aalto University.