INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Contemporary Polish Language Model (Version 2) Using Big Data and Sub-Word Approach

Krzysztof Wołk

Polish-Japanese Academy of Information Technology
kwolk@pja.edu.pl

Abstract

Language and vocabulary continue to evolve in this era of big data, making language modelling an important language processing task that benefits from the enormous data in different languages provided by web-based corpora. In this paper, we present a set of 6-gram language models based on a big-data training of the contemporary Polish language, using the Common Crawl corpus (a compilation of over 3.25 billion webpages) and other resources. The corpus is provided in different combinations of POS-tagged, grammatical groups-tagged, and sub-word-divided versions of raw corpora and trained models. The dictionary of contemporary Polish was updated and presented, and we used the KENLM toolkit to train big-data language models in ARPA format. Additionally, we have provided pre-trained vector models. The language model was trained, and the advances in BLEU score were obtained in MT systems along with the perplexity values, utilizing our models. The superiority of our model over Google's WEB1T n-gram counts and the first version of our model was demonstrated through experiments, and the results illustrated that it guarantees improved quality in perplexity and machine translation. Our models can be applied in several natural language processing tasks and several scientific interdisciplinary fields.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

There are several language processing tasks for which web-scale corpora are required, primary because they contain massive amounts of information in various languages. One crucial task is language modelling, and web-scale language models have proven effective in improving the recognition performance of automated speech and quality of machine translation [1-3]. Some other natural language processing (NLP) tasks also rely heavily on language modelling, for example, language quantification, automatic speech recognition, etc. [4]

Some language models are qualified based on the Common Crawl corpus and n-gram counts. Google released n-gram counts that were trained on 1 trillion text tokens [5]. N-grams, which occurred in less than 40 instances, were clipped, and the words that occurred less than 200 times were each substituted with an unknown word. The clipping makes the counts unsuitable for performing an evaluation of a language model using the Kneser-Ney smoothing algorithm as the algorithm requires unclipped counts, although clipping occurs in the last model.

There is another trial related to the Google n-gram counts that is publicly available [5] as the training information was not de-duplicated. In other words, boilerplate, similar to copyright notices, have extremely high counts [6]. Although Google shared its version [7] in a restricted context [6] that was subject to de-duplication, it was never formally issued to the public [8]. Before the n-grams were added, the data for training were subject to de-duplication. Microsoft provides a web service [9] for making queries in terms of language model probabilities. However, this service applies only the English language, whereas our methodology on model preparation is compatible with more languages. Furthermore, an experiment [9] was conducted on the re-ranking of machine translations of the Polish language as the service crashed a few times owing to the number of output queries created, even with client-side caching. Using the Microsoft service for the entire machine translation decoding would imply a prerequisite for low latency and large queries.

In our previous work, a big-data model of the Polish language based on the Common Crawl repository was created and made available. We had introduced a 5-gram model, which was trained and made available in the ARPA format and in the form of pure text data. The model was algorithmically divided into sentences and de-duplicated, and the data were cleaned because the Common Crawl corpus is extremely noisy. Originally, the Polish corpus was 296 GB in size and comprised 1,962,047,863 sentences. After cleaning, it became 94 GB and comprised 920,517,413 sentences. Despite this reduction, the model achieved better results in terms of machine translation quality and perplexity as compared to WEB1T. The corpus was awarded at the LTA conference in 2017 [11].

Over time, the amount of available data has increased with the evolution of language and vocabulary. Therefore, we decided to update our previous language model. The Common Crawl corpus [12] was retrieved this time, considering the data created in 2017, 2018, and 2019. As before, the data was de-duplicated and normalized, cleaned of tags and other unnecessary information, and rid of fragments written in languages other than Polish. Additionally, coding problems were eliminated, and tokenization was performed. The corpus was divided into sub-word units and annotated with additional grammatical information. Some tools [13, 14] were employed for this purpose.

This present study shows the way to build a contemporary language model from big text data for any language supported in the Common Crawl project (based on the Polish language). The quality of our model was compared to its previous version [11] and the Google WEB1T [10] model. Prior to this comparison, we performed the quality evaluation of our new approach by measuring perplexity and illustrating the improved quality of machine translation systems using the new model. Finally, we made the results of our work publicly available as plain text data, plan text data divided into sub-word units with different methods, plain text data annotated with different

http://dx.doi.org/10.21437/Interspeech.2020-1207

methods, trained 6-gram ARPA language models [15], pre-trained vector models [16,17], a dictionary organized according to the most recurrent unigrams, and a dictionary cleaned from numbers, names. and less likely words. The data can be obtained free of cost at https://tinyurl.com/biglmv2. Additionally, our vector models were accepted as part of the NLPL word embeddings repository at http://vectors.nlpl.eu/repository [18].

2. Data preparation

The pre-processing step involved solving numerous problems encountered with the data. The first problem was that of data selection in a single language. Common Crawl also has some encoding errors while parsing to UTF-8 and therefore led to spelling errors. In addition, some texts were repeated numerous times, e.g., “copyright,” “comment,” “data,” etc. Several text structures were ungrammatical or included odd insertions. Certain language-specific difficulties were also encountered that needed to be addressed separately for each language. Additionally, the data covered samples of spoken texts, such as dialogs, written articles, and literature. It was also impossible to define the text domain.

The early stage of the data acquisition pipeline was used to separate the information according to language. We considered the possibility of an automatic detection of the main language for each page. However, we found that the mixed language commonly occurred within one page. A Python tool has been implemented, and it worked in three phases. Initially, the Python LangDetect [20] library was used to find whole pages that appeared to be in Polish. In the second phase, plWordnet [21] was used for the vocabulary comparison of extracted articles using Polish vocabulary. Articles that included less than 30% of Polish words were removed. Furthermore, the aspell tool was employed before using the plWordnet to correct spelling errors, which made automatic correction possible. In the last step, the text was divided into sentences using an automatic tool [22]. This technique made it possible for us to collect 732 GB of pure textual data. The text consisted of a total of 4,944,846,573 sentences.

It is crucial to remove repetitive data as they can affect the statistical model. Notably, there is a frequent repetition of some texts on the internet, for example, press information. To decrease such volume, all lines that were duplicated were removed using implemented tool. Comparison was performed at the sentence level. Detailed information of the data quantity before and after deduplication are presented in Table 1.

Table 1: *De-duplication and cleaning results.*

	Size in GB	Number of sentences	Number of unique tokens
Before	732	4,944,846,573	567,483,294
After	196	1,908,582,538	421,344,934

The de-duplication and cleaning steps removed approximately 75% of the Polish data in terms of tokens. This is comparable to the reductions reported by Bergsma et al. [23].

In addition to the de-duplication, the data were limited to printable UTF-8 characters, all email addresses were replaced with the same address, and the left-over HTML tags were removed. Before creating the language models, we normalized the punctuation using the script provided by the WMT[24]. Tokenization was performed using the Moses tokenizer [25], which was followed by the application of the Moses true caser [25].

2.1. Sub-word units and annotation

Owing to rich morphology of the Polish language, data pre-processing was necessary to reduce the vocabulary. In many current applications associated with NLP, especially for morphologically complex languages, a limited and closed dictionary is used. The use of this dictionary not only limits the solution functionality, but also introduces computational limits and forces frequent system trainings in a dynamically changing language. The solution to these problems is the use of so-called “open dictionary,” which feature units that are smaller than words for all or part of the words in the text. The byte pair encoding (BPE) technique [26] is commonly used for English, and an equivalent for Polish is proposed in this paper. The proposed tool enables the division of text via two methods: according to the syllables and core with suffixes and prefixes, following given rules. The user can automatically annotate divisions with different tags. By default, the "++ – –" symbol is appended, thereby recording how—and from which side—given units connect to each other to form a word [27]. The proposed solution was found to function more effectively than full word forms and BPE. Our tool can also tag divided text with parts-of-speech (POS) and tags from 255 grammatical groups.

3. Methods

3.1. Evaluation

We used the perplexity measure to measure the performance of our new language model [28-31]. Three models were compared in this study: WEB1T, the first version of our model [11], and this present (second) version. Adding the data created between 2017 and 2019 and indexed in the Common Crawl project, a new, larger 732 GB model with 4,944,846,573 sentences was obtained. Additionally, the new model was a 6-gram model after training, instead of the 5-gram model of the first version. Details on the number of n-grams are presented in Table 2.

Table 2: *Test model specification for Polish.*

	COMMON	COMMON v2
Size	296 GB	732 GB
Sentences	1,962,047,863	4,944,846,573
Cleaned	94 GB	196 GB
Cleaned	920,517,413	1,908,582,538

Furthermore, employing the datasets used in [11], the Moses SMT toolkit was used to train three statistical machine translation models. The translation order was English-to-Polish. We enriched the translation systems using the prepared language models and evaluated them using BLEU [32] metric. As much as possible, we ensured that the new experiments were conducted in the same environment as in [11].

The baseline results of SMT systems for each corpus are presented in Table 3. Tree different test sets were selected from a corpus of TED lectures from the IWSLT conference, European Medicines Agency Leaflets (EMEA) corpus, and OpenSubtitles corpus. From these three corpora, 1,000 sentences were randomly selected for assessment using perplexity.

Table 3: *Baseline system results.*

Corpus name	Baseline system score (BLEU)
TED	17.42
EMEA	36.74
OPEN	58.52

We used the KENLM toolkit for the language model [33] training. This tool has been used for training 6-gram language models.

In the case of machine translation, the experiment management system [25] was used from the opensource Moses SMT toolkit to perform the experiments. SyMGIZA++ [34] was used. The OOV's were handled using the unsupervised transliteration model [35].

To summarize this study, we used the first version of the big data Common Crawl-based corpus (CCv1) [11], Google corpus (WEB1T), and the second version of COMMON (CCv2). Details of the corpora and number of n-grams are presented in Table 4.

Table 4: *Number of N-grams in language models.*

	CC v1	WEB1T	CC v2
1-grams	102,742,823	9,749,397	421,344,934
2-grams	1,227,434,111	72,096,704	2,978,853,249
3-grams	1,208,818,561	128,491,454	2,003,857,026
4-grams	1,513,980,357	128,789,635	2,270,689,390
5-grams	1,433,864,427	113,097,133	2,088,765,597
6-grams	n/a	n/a	1,844,154,756

3.2. Experiments

The experiments with the TED lectures [19], OPEN [36], and EMEA [37] corpora examined the perplexity of the data. We prepared several types of LMs for this evaluation, which were as follows:

- 6-gram closed vocabulary LM based on full word forms (RAW)
- 6-gram closed vocabulary LM based on full word forms with POS tags (RAW_POS)
- 6-gram closed vocabulary LM based on full word forms grammatical groups tags (RAW_GR)
- 6-gram open vocabulary LM based on Byte Pair Encoding algorithm (BPE)
- 6-gram open vocabulary LM based on stemming algorithm (STEM)
- 6-gram open vocabulary LM based on stemming algorithm followed by BPE algorithms (STEM_BPE)
- 6-gram open vocabulary LM based on stemming algorithm with POS tags (STEM_POS)
- 6-gram open vocabulary LM based on syllables (SYL)
- CBOW vector model based on FastText (FTC)
- Skip-gram vector model based on FastText (FTS)
- CBOW vector model based on Word2Vec (WVC)
- Skip-gram vector model based on Word2Vec (WVS)

For the vector models as we used the Gensim library [38], to calculate the perplexity of those models, we first to retrieved the loss by passing the compute_loss=True parameter gensim.models.word2vec.Word2Vec constructor. This way, we stored the loss while training. Once trained, we called the get_latest_training_loss() method to retrieve the loss. Owing to the loss in the cross-entropy loss of the model, the perplexity was obtained by raising 2 to the power of the loss (2**loss) for the vector models [39].

Additionally, three baseline systems (baseline BLEU) were trained, and we augmented them with our language models based on Common Crawl. We acted accordingly while using WEB1T language model. The translation was performed into Polish.

Table 6: *Perplexity-based language model evaluation.*

Corpus	Model	Perplexity
TED	CC v1	1471
	WEB1T	1523
	RAW	1409
	RAW_POS	1390
	RAW_GR	1418
	BPE	1401
	STEM	1363
	STEM_BPE	1359
	STEM_POS	1354
	SYL	1355
	FTC	1338
	FTS	1341
	WVC	1369
	WVS	1372
OPEN	CC v1	480
	WEB1T	671
	RAW	469
	RAW_POS	462
	RAW_GR	464
	BPE	434
	STEM	423
	SMTEM_BPE	419
	STEM_POS	428
	SYL	473
	FTC	452
	FTS	461
	WVC	443
	WVS	438
EMEA	CC v1	1163
	WEB1T	1253
	RAW	1145
	RAW_POS	1132
	RAW_GR	1136
	BPE	1147
	STEM	1114
	SMTEM_BPE	1136
	STEM_POS	1144
	SYL	1106
	FTC	1095
	FTS	1076
	WVC	1074
	WVS	1059

4. Results

The perplexities of the test sets are presented in Table 6. In the table, the previous version of our model is mentioned as CCv1 and Google's model as WEB1T.

The findings derived from our language model evaluation by means of SMT systems are presented in Table 7. The "Delta" column in the table refers to the difference between the baseline and augmented systems. It should be observed that no in-domain adaptation of the language models were conducted. As we wanted to recreate the exact experimental environment in [11], we chose the Moses SMT system that is not compatible with vector models; thus, they were not tested in MT.

Table 7: *SMT-based language model evaluation.*

Corpus	Language model	Baseline BLEU	Augmented BLEU	Delta
TED	CC v1	17.42	18.33	0.91
	WEB1T	17.42	17.97	0.55
	RAW	17.42	18.79	1.37
	RAW_POS	17.42	19.01	1.59
	RAW_GR	17.42	18.85	1.43
	BPE	17.42	19.23	1.81
	STEM	17.42	19.46	2.04
	SMTEM_BPE	17.42	19.37	1.95
	STEM_POS	17.42	19.76	2.34
	SYL	17.42	18.13	0.71
OPEN	CC v1	58.52	59.23	0.71
	WEB1T	58.52	59.01	0.49
	RAW	58.52	59.94	1.42
	RAW_POS	58.52	61.34	2.82
	RAW_GR	58.52	60.34	1.82
	BPE	58.52	60.75	2.23
	STEM	58.52	60.87	2.35
	SMTEM_BPE	58.52	61.63	3.11
	STEM_POS	58.52	61.57	3.05
	SYL	58.52	60.08	1.56
EMEA	CC v1	36.74	38.34	1.6
	WEB1T	36.74	37.93	1.19
	RAW	36.74	38.92	2.18
	RAW_POS	36.74	39.24	2.50
	RAW_GR	36.74	39.16	2.42
	BPE	36.74	39.78	3.04
	STEM	36.74	39.95	3.21
	SMTEM_BPE	36.74	40.25	3.51
	STEM_POS	36.74	40.08	3.34
	SYL	36.74	38.43	1.69

5. Discussion and conclusions

In this study, we effectively released several types of 6-gram counts and built language models using big-data textual corpora. These models overcame the restrictions of other smaller, publicly available resources. Additionally, we provided four types of vector models and made all our data publicly available at https://tinyurl.com/biglmv2. Additionally, our vector models were accepted as part of the NLPL word embeddings repository at http://vectors.nlpl.eu/repository [18].

From the results of our experiments, we observed that vector and sub-word-based open vocabulary models were highly effective. This is consistent with other findings reported in the literature [40]. Furthermore, we could illustrate that after data pre-processing, the result for BLEU and perplexity results outperformed those of the state-of-the-art language models, such as COMMON [11] and WEB1T. We further observed that syllables seemed too small as units compared to stemming and BPE. On the contrary, both BPE and stemming provided better-performing systems than our baselines, but the best scorers were those that combined both stemming and BPE. Adding additional lexical information such as POS to stems also proved effective. Future studies may investigate the combination of BPE, stemming, and POS in a single corpus.

Furthermore, smaller corpora (e.g., Opus Project [41], Wikipedia [42], etc.), even after merging, are smaller than the amount of data used in this study. We proved that the enhancement of perplexity and machine translation provide a better utilization of language knowledge. The results of our work are publicly available for free. What we shared are the raw data after pre-processing, raw data tagged with POS or grammatical groups, raw data divided into sub-word units of different types (syllables, stemming, BPE), and sub-word-divided raw data with POS. In addition, we trained and shared 6-gram language models with pruned 20% of less likely n-grams for all raw datasets, vector models, a dictionary with the number of the most common Polish words based on the Common Crawl corpus, and a dictionary without the numbers, which was manually cleaned from noisy data by native Polish translators. Our models have limitless commercial and scientific applications. We believe that our models can be applied not only to various NLP tasks, but also to other fields of science, especially the interdisciplinary fields (e.g., computational linguistics, digital humanities, ASR, MT, language quantification, phraseological competence analysis, etc.) [42].

6. References

[1] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 858-867, 2007.

[2] D. Guthrie and M. Hepple, "Storing the web in memory: Space efficient language models with constant time retrieval," *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, October*, *Association for Computational Linguistics,* pp. 262-272, 2010.

[3] C. Chelba and J. Schalkwyk, "Empirical exploration of language modeling for the google.com query stream as applied to mobile voice search," in *Mobile Speech and Advanced Natural Language Solutions, Springer New York, 2013,* pp. 197-229, DOI: 10.1007/978-1-4614-6018-3_8

[4] A. Lenko-Szymanska, "A corpus-based analysis of the development of phraseological competence in EFL learners using the CollGram profile, in "*The 7 th Conference of the Formulaic Language Research Network (FLaRN), Vilnius, 28-30 June*, 2016, pp. 28-30.

[5] T. Brants and A. Franz, "Web 1T 5-gram corpus version 1.1.," *Google Inc., Spetember, 2006.*

[6] D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, K. Dalwani, and S. Narsale, "Unsupervised acquisition of lexical knowledge from n-grams: Final report of the 2009 JHU CLSP workshop," altimore: John Hopkins University," 2010, Available at< http://www. clsp. jhu. edu/vfsrv/workshops/ws09/documents/Lin. pdf, 2010.

[7] S., Bergsma, E. Pitler, and D. Lin, "Creating robust supervised classifiers via web-scale N-gram data" *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.*

Association for Computational Linguistics, July, pp. 865-874, 2010.
[8] D. Lin, Personal communication, October, 2013.
[9] K. Wang, C. Thrasher, E. Viegas, X. Li, and B.J.P. Hsu, "An overview of Microsoft Web N-gram corpus and applications," *Proceedings of the NAACL HLT 2010 Demonstration Session. Association for Computational Linguistics*, June 2, pp. 45-48, 2010.
[10] K. Wróbel, "Plujagh at semeval-2016 task 11: Simple system for complex word identification," *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), June,* pp. 953-957, 2016.
[11] K. Wołk, A. Wołk, and K. Marasek, "Big data language model of contemporary Polish," in *2017 Federated Conference on Computer Science and Information Systems (FEDCSIS), September, 2017, IEEE*, pp. 389-395
[12] S. Roziewski, W. Stokowiec, and A. Sobkowicz, "N-gram collection from a large-scale corpus of polish internet," in *Machine Intelligence and Big Data in Industry*, 2016, pp. 23-34.
[13] K. Wołk, E. Zawadzka, and A. Wołk, "Statistical approach to noisy-parallel and comparable corpora filtering for the extraction of bi-lingual equivalent data at sentence-Level," in *World Conference on Information Systems and Technologies, March, 2018,* pp. 797-812.
[14] S. Krauwer, and E. Hinrichs, "The CLARIN research infrastructure: resources and tools for e-humanities scholars" *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014) European Language Resources Association (ELRA),* pp. 1525-1531, 2014.
[15] J. Kadivec, M. Robnik-Šikonja and S. Vintar, "ccGigafida ARPA language model 1.0.," 2017.
[16] J. Misztal-Radecka, "Building semantic user profile for polish web news portal," *Computer Science*, vol. 19, 307-332, 2018.
[17] J. Kocoń and M. Gawor, "Evaluating KGR10 Polish word embeddings in the recognition of temporal expressions using BiLSTM-CRF," Schedae Informaticae, 2018(Volume 27).
[18] M. Fares, A. Kutuzov, S. Oepen, & E. Velldal, "Word vectors, reuse, and replicability: Towards a community repository of large-text resources", *In Jörg Tiedemann (ed.), Proceedings of the 21st Nordic Conference on Computational Linguistics*, NoDaLiDa, Linköping University Electronic Press, 2017.
[19] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT evaluation campaign," *Proceedings of the International Workshop on Spoken Language Translation, Heidelberg, Germany, December,* 2013.
[20] Language detection library ported from Google's languagedetection. https://pypi.python.org/pypi/langdetect?
[21] M. Maziarz, M. Piasecki, and S. Szpakowicz, "Approaching plWordNet 2.0" *Proceedings of 6th International Global Wordnet Conference, The Global WordNet Association,* pp. 189-196, 2012.
[22] K. Wołk and K. Marasek, "Polish – English speech statistical machine translation systems for the IWSLT 2014," *Proceedings of the 11th International Workshop on Spoken Language Translation, Tahoe Lake, USA,* pp. 143- 149, 2014.
[23] S. Bergsma, E. Pitler and D. Lin, "Creating robust supervised classifiers via web-scale N-gram data," *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, July,* pp. 865-874, 2010.
[24] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M., Post, R., Soricut and L. Specia, "Findings of the 2013 workshop on statistical machine translation," *Proceedings of the Eighth Workshop on Statistical Machine Translation, Sofia, Bulgaria. Association for Computational Linguistics,* pp. 1-44, 2013.
[25] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, ... and C. Dyer, "Moses: Open source toolkit for statistical machine translation," *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions. Association for Computational Linguistics, June,* pp. 177-180, 2007.
[26] R. Sennrich, B. Haddow and A. Birch, "Neural machine translation of rare words with subword units," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, August,* vol. 1: long papers, pp. 1715-1725, 2016.
[27] K. Wolk and K. Marasek, "Survey on neural machine translation into Polis," in *International Conference on Multimedia and Network Information System, Wroclaw, Poland, September 12-14, Springer, Cham, 2018*, pp. 260–272.
[28] S. F. Chen, and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Proceedings of the 34th annual meeting on Association for Computational Linguistics Association for Computational Linguistics*, *June,* pp. 310-231, 1996. DOI: 10.3115/981863.981904
[29] Perplexity [Online], "Hidden Markov model toolkit website," Cambridge University Engineering Dept. Available: http://www1.icsi.berkeley.edu/Speech/docs/HTKBook3.2/node188_mn.html, retrieved on March 29, 2020.
[30] P. Koehn, "Moses, statistical machine translation system, user manual and code guide," 2010.
[31] D. Jurafsky, [Online], "Language modeling: Introduction to ngrams," Stanford University. Available: https://web.stanford.edu/class/cs124/lec/languagemodeling. pdf, etrieved on November 29, 2015.
[32] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," *Proceedings of the 40th annual meeting on association for computational linguistics Association for Computational Linguistics, July,* pp. 311-318, 2002.
[33] K. Heafield, "Kenlm language model toolkit," *Kenlm. Code. Kenneth Heafield. Np, nd Web, 3*, 2015.
[34] M. Junczys-Dowmunt and A. Szał, "Symgiza++: symmetrized word alignment models for statistical machine translation," in *Security and Intelligent Information Systems. Springer Berlin Heidelberg*, pp. 379-390, 2012. DOI: 10.1007/978-3-642-25261-7_30
[35] N. Durrani, H. Sajjad, H. Hoang and P. Koehn, "Integrating an unsupervised transliteration model into statistical machine translation," in *EACL, April,* vol. 14, pp. 148-153, 2014. DOI: 10.3115/v1/E14-4029
[36] M. Müller and M. Volk, "Statistical machine translation of subtitles: From OpenSubtitles to TED," in *Language processing and knowledge in the Web. Springer, Berlin, Heidelberg,* pp. 132-138, 2013.
[37] M. Neves, A. J. Yepes and A. Névéol, "The scielo corpus: a parallel corpus of scientific publications for biomedicine," *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), May,* pp. 2942-2948, 2016.
[38] B. Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, spaCy, and Keras*, Packt Publishing Ltd., 2018.
[39] M. A. Kharazmi and M. Z. Kharazmi, "Text coherence new method using word2vec sentence vectors and most likely n-grams," in *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, *IEEE, December*, pp. 105-109, 2017.
[40] S. J. Mielke and J. Eisner, "Spell once, summon anywhere: A two-level open-vocabulary language model," *Proceedings of the AAAI Conference on Artificial Intelligence, July,* vol. 33, pp. 6843-6850), 2019.
[41] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Lrec, May 23,* vol. 2012, pp. 2214-2218, 2012.
[42] T. Baumann, A. Köhn and F. Hennig, "The spoken Wikipedia corpus collection: Harvesting, alignment and an application to hyperlistening." *Language Resources and Evaluation*, vol. 53, no. 2, pp. 303-329, 2019.
[43] A. Clark, C. Fox and S. Lappin, (Eds.) *The Handbook of Computational Linguistics and Natural Language Processing*, John Wiley & Sons, 2013.