This is a post-peer-review, pre-copyedit version of an article published in 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA). The final authenticated version is available online at: https://doi.org/10.1109/INISTA.2017.8001124

Towards Facts Extraction from Text in Polish Language

Tomasz Boiński and Adam Chojnowski
Department of Computer Architecture
Faculty of Electronics, Telecommunication and Informatics
Gdańsk University of Technology
11/12 Narutowicza Street
80-233 Gdańsk, Poland
Email: tobo@eti.pg.gda.pl

***Abstract*—Natural Language Processing (NLP) finds many usages in different fields of endeavor. Many tools exists allowing analysis of English language. For Polish language the situation is different as the language itself is more complicated. In this paper we show differences between NLP of Polish and English language. Existing solutions are presented and TEAMS software for facts extraction is described. The paper shows also evaluation of the proposed solution and the tools used. Finally some conclusions are given.**

***Index Terms*—Natural Language Processing, Facts extraction, Polish natural language**

Acknowledgment

Copyrights: IEEE

The original paper is available as: Boiński, Tomasz, and Adam Chojnowski. "Towards facts extraction from text in Polish language." INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on. IEEE, 2017, https://doi.org/10.1109/INISTA.2017.8001124

I. Introduction

Natural Language Processing (NLP) finds many usages in different fields of endeavor. Growing number of applications try to communicate with the user using natural language. Usually this process is based on some kind of knowledge base containing facts and relations between them stored in a manner possible for computer processing. There is a need for creation of such databases. They can be defined manually but this process is very costly and time consuming. Furthermore in many fields communication using natural language is still a primary form of knowledge exchange [1]. Thus a need for elaborate tools for fact extraction arises.

A the beginning NLP was based solely on preprogrammed grammatical rules and dictionaries. This approach did not give satisfactory results so an augmented transition network approach was proposed [2] which utilized general knowledge in facts extraction. In the 80thies statistical and corpora approaches emerged [3]. This approach proved to give good results and is currently widely used [4].

For English language, which is rather simple, the situation looks quite promising. Many interesting solutions exists that cope with the task in a satisfactory manner [5], [6], [7]. Solutions like NELL [8], FRED [9] or TIPALO [10] allows task extraction from text and their verification.

For Polish language the situation is rather different. With complex grammar and many irregularities the language is difficult for automated processing. With new tools emerging it is however possible to create at least semi-automatic solution that will help in extracting facts from Polish texts.

In this paper we try to look on the most commonly used tools for sentence analysis for Polish language and evaluate them using proposed TEAMS system for extracting facts from Polish text.

The structure of the paper is as follows. Section II shows differences between NLP of Polish and English language, later description of current solutions is given and in section V the TEAMS evaluation tool is described. Section VI presents evaluation of the proposed solution and the tools used. Finally some conclusions are given.

II. Natural Language Processing in Polish Language

Analysis of Polish language is more complex than English due to morphological and syntactical differences between those two languages. English language is an isolating language (with minor exceptions) while Polish is an inflexive language. This imposes that the words seldom are composed of single morfems and they can take many forms due to declination of conjugation. Another difference is that English language is positional while Polish is casual. That means that the meaning of the sentence is not dependent on the order of the words rather on their form.

In general in English language a parser analyzing a word usually gets its base form and the grammatical role of the word in the sentence can be determined on current position in the sentence. In Polish language the parser analyzing a word needs to find given words both substantive and grammar meaning. Those to meanings depends on each other. Lets consider a simple example: “Kot je mysz” (“Cat eats mouse”). In English grammatical role of each word is obvious as it can be directly derived from subject-predicate-object

construction of the sentence. In Polish however the case is more complicated, The word “mysz” (“mouse”) can be both accusative and denominator. In this case the parser can by reduction deduce that it is an accusative as the word “kot” (“cat”) is a denominator and verb “je” (“eat”) requires the complementation using the accusative case. In plural the case becomes even more difficult. In a sentence “Koty jedzą myszy” (“Cats eat mice”) both words cats and mice can be treated as denominator. We can assume in this case subject-predicate-object order but this is not the case in many situations. Simple grammatical analysis is thus prone to errors. Furthermore in Polish language different types of inflection of the word are syncretic.

There are some rules describing (among many others) how the words should be ordered within a sentence [11] or how attributes should be attached to a word [12]. Those rules are often implemented within parsers but many texts are of low quality regarding the style of the writing and those rules might not apply.

The above example shows that the complexity of Polish language, combined with limited impact of the language when compared with English language, makes availability of tools for facts extraction scarce and of lower quality that those for English language.

III. Towards Facts Extraction

Having proper grammatical analysis of a sentence one can start extracting facts from the text. In our work we aim at automatic facts extraction from any type of texts. We can distinguish four approaches [13], [14]:

- extraction based on the text structure – the parser needs to identify the concepts in the text (usually named entities) and their coreferences and apply them to templates to find relations within the text;
- extraction based on predefined ontology – a predefined ontology with classes and relations exists and is used as a template for finding facts within text;
- extraction based on the ontology created from analyzed texts – the ontology is created on the fly based on the information found within the text, this ontology than supports further facts extraction;
- semantic annotation – external knowledge base is used for finding and annotating concepts in analyzed texts, this way its easier to find relations and the meaning of the word.

IV. State of the Art

For English language many solution exists that either is designed directly for facts extraction or can be used in the process. They used different techniques and are characterized by varying result quality. NELL (Never Ending Language Learner) [8] is a self-learning tool for facts extraction from text corpora based on automatically created rules. In 2015 NELL had around 50 million facts, including 2 million facts regarded as of high reliability. FRED [9] is based on multiple techniques and provides results in form of rdf/owl triples representing dependencies in the text. During facts extraction it performs syntactic analysis, named entities recognition, dependency detection and meaning of the homographs. TIPALO, an extension to FRED, enriches FRED results with relations of type “x belongs to y”, “x is synonym of y” and with connections to WordNet and DBpedia. This tool however is used only to parse Wikipedia and takes into account only the first paragraph on each page.

Probably the most notable solution for English language is the IBM Watson supercomputer [15], [16]. The system can ask and respond to questions formulated in natural English language. System itself is split into set of designated algorithms cooperating together to provide the best possible answer for given question. Data is retrieved from all around the Web in form of unstructured text which is then analyzed semantically and stored internally for future use. IBM claims that Watson is constantly updating its knowledge resources every day with new information. Deep learning techniques are then used to prepare system for even quicker and more accurate data retrieval possibilities. When input question is provided Watson utilizes dozens of techniques of information retrieval which provide candidate answers ranked by their relevance and accuracy in form of feedback-loop, where every step of processing could be done multiple times for the best possible results.

For Polish language few solutions also exists but mainly operating at the level of part of speech annotation. Świgra [17] is a parser written in Prolog that can create a syntactic tree for the analyzed sentence. It is based on a simplified formal Polish grammar [18] and Polish language dictionary, taking into account abnormalities specific to the Polish language. Similar tool is a TaKipi [19] tagger. It utilizes Morfeusz software library [20] for morfo-syntactic analysis and a set of semi-automatically defined rules. General effectiveness of the tagger is around 93%, when taken into account only sentences with at least two possible interpretation the effectiveness drops to around 80% [19].

The Nekst project realized by Polish Academy of Sciences with cooperation from Wrocław University of Technology [21] aimed at creation of intelligent search engine for Polish Web resources. Algorithms and solutions developed for purposes of this project are oriented around contextual analysis of text resources in terms of facts, relations, sentimental bias, hierarchical ontologies and massive parallelisation of aforementioned mechanisms. Unfortunately, after losing financing from European Union product seems like no longer developed. Newest indexed documents corresponds to scheduled end of support, however it seems like several major achievements in terms of processing Polish texts were concluded.

An interesting solution is Multiservice [22] run by Institute of Computer Science Polish Academy of Science. It allows usage of different tools for processing of Polish language and creating a chain of calls that transform the result of the previous call. We chose this service as a base for our evaluation as it uses most of the tools available for Polish language.

V. TEAMS – Truth Extraction and Maintenance Service

Our proposed TEAMS application is similar to NELL/TIPALO services however working on Polish language. Our goal was also to use as much preexisting solutions as possible so it can serve as an evaluation tool.

The task of TEAMS software is to create a knowledge base from the raw text given by the user or from a Wikipedia articles. In general the text is parsed using proper functions from Multiservice and the results are matched with subject-predicate-object template. The knowledge base produced is given in form of triples and can contain all relations or only the set matching a template given by the user. The concepts and relations in the knowledge base are in their base form and contain information about the negation, reflexivity etc.

The TEAMS software is designed in a modular fashion for ease of modification and extendability. The most notable modules are:

- wikiPreparser – a module that allows analysis of Wikipedia pages. We analyze whole pages but we need to strip it from HTML tags and divide the text into fragments suitable for Multiservice (currently no more than 5000 characters). It is worth mentioning that altho we look on the whole text, the tagger only works within a scope of one sentence. This module also needs to transform the text to some extend – the Multiservice does not understand characters like "–" so it needs to be replace with proper text (in this case with "is a").
- multiserviceCall – a module that communicates with the Multiservice web site and runs the following tools: Concraft [23], Spejd [24], Nerf [25], Mention Detector [26] and Dependency Parser [27].
- mentionsHierarchy – groups equal subjects from different sentences returned by the Multiservice.
- factBaseMaintainer – a module that simplifies the knowledge base by transforming all concepts to their base form and a single synonym from all available versions. This step is based on the Słowosieć [28], the Polish WordNet.
- relationsParser – the main module of the system. Based on the results form Multiservice the tool detects relations between the facts found in the sentences. The algorithm goes as follows:
  - the sentences with no subjects are discarded,
  - in the remaining sentences predicates are located,
  - for each predicates group of words that can be either subjects or objects are detected,
  - if there are no subjects and/or objects in the sentence than as a subject we take the nouns located directly before the predicate and as objects the ones directly after the predicate.
  - if there is no subject and the tool is working on a Wikipedia article the first subject within the article is treated as default subject.

The communication workflow between the modules is shown in Fig 1. The user, in this case, gives link to the Wikipedia article. The Page is then downloaded and parsed by wikiPreparser module. The raw text is forwarded to Multiservice via multiserviceCall module and analyzed using Concraft, Spejd, Nerf, Mention Detector and Dependency Parser tools (in the given order). TEAMS software analyzes the results using mentionsHierarchy module and finally for each sentence using relationsParser potential relations are discovered. The resulting triple set is passed to the factBaseMaintainer, which unifies synonyms using Słowosieć.

VI. Evaluation

To evaluate our solution and the external tools test using Wikipedia articles were performed. Pages containing information from two different fields were used:

- exact science: Polip (eng. *polyp*) [29], Knidoblasty (eng. *cnidoblasts*) [30], Kostkowce (eng. *box jellyfish*) [31], Meduza (eng. *medusa*) [32], Ukwiały (eng. *anemones*) [33].
- humanities (often using more descriptive language): Wojciech Jaruzelski [34], Lech Wałęsa [35], Aleksander Kwaśniewski [36], Lech Kaczyński [37], Bronisław Komorowski [38], Andrzej Duda [39].

During the analysis only the text was parsed, all tables, enumerations and data from info-boxes were removed. The facts are detected based on their role in the sentence. The facts have to match the subject-predicate-object pattern. The predicates also include negation and the direction of the relations. The results, as presented in Table I are far from satisfactory. In some cases, usually when the articles were short and written in compact form, the accuracy was quite high. In case of article about cnidoblasts the amount of correctly detected facts is high (around 80%). The facts detected in this case are presented in Table II (in Polish). The manual fact extraction was done in the same way as the automatic one - the concepts were identified within the text based on their role in the sentence (usually subjects) and connected with objects suing found predicates.

TABLE I
The results from running TEAMS on the evaluation set

Not detected facts	Discarded by TEAMS due to the location (tables, info-boxes etc.)	11%
	Missing object	3%
	Not marked by Multiservice	48%
	Total	**62%**
Facts detected incorrectly	Marked incorrectly by Multiservice	26%
	Incorrectly analyzed by TEAMS	3%
	Total	**29%**
Correctly detected facts	**Total**	**9%**

The errors were introduced in many cases by the Multiservice analysis, mainly by dependency parser. Despite boasting with high accuracy (over 80%) the system had many difficulties with complex sentences with multiple subjects and the containing many insertions, parentheses etc. Also when there was no explicit subject or the subject was a pronoun

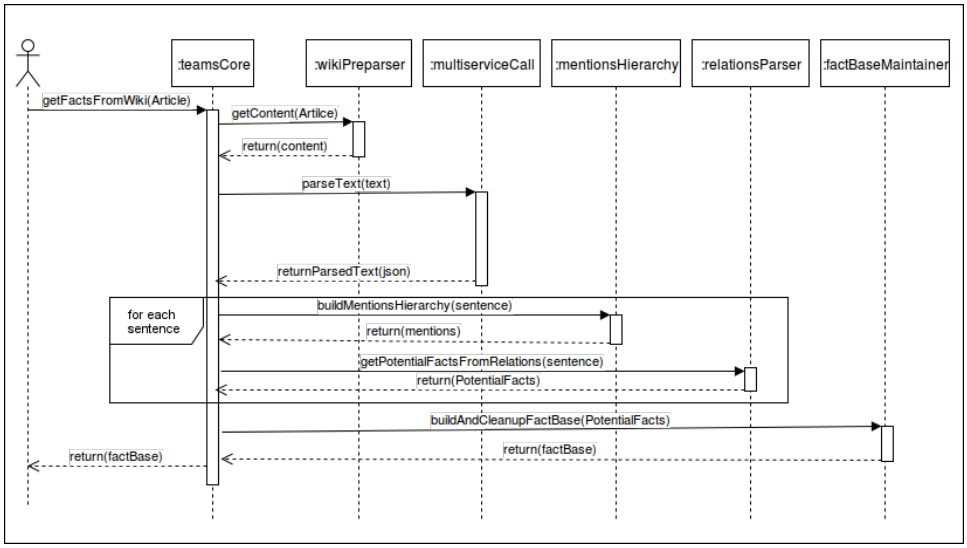

Fig. 1. Communication workflow in the TEAMS solution

TABLE II
FACTS GATHERED FROM THE ARTICLE ABOUT CNIDOBLASTS (IN POLISH)

Subject	**Predicate**	**Object**
knidoblast	być	typ
komórka+parzydełkowy	powstawać+z	komórka+macierzysty
knidoblast	powstawać+z	komórka+interstycjalny
obecność+knidoblast	być	cecha+parzydełkowiec

the dependency parser introduced many errors. An example of this case is the sentence "W czerwcu 1961 ukończył tę szkołę (...)" (eng. "He finished this school in June 1961 (...)") from the article about Lech Wałęsa. In this case the month and year (June 1961) were marked as a subject in the sentence.

Many omitted facts were stored in tabular environments within the Wikipedia pages and as such were discarded. Those fragments should be analyzed separately as, by not being a proper sentence, they break the behavior of Multiservice thus increasing amount of incorrectly generated facts. However separate analysis of tabular environments could provide good source of facts as such data is already structured very well.

The remaining errors are related to the TEAMS software itself. In case of complex sentences, even if they are analyzed properly by the Multiservice, as TEAMS is dependent on the order of occurrence of the elements in the sentence. In most cases such approach proved to limit the error ratio but still can introduce errors even if the sentence is correctly tagged using Multiservice. Also the decision how to treat cases where there is no subject or object detected influences error ratio. We decided to try re-detecting such concepts by the order of appearance in the text. This can increase the error level as some facts are generated incorrectly. However without this mechanisms the number of not detected facts grew even further.

During our tests we also came up upon one very interesting case. The case is related to w verb "mają" (eng. having). In Polish language it can be treated as a form of verb "maić" which is a synonym to "zazielenia/'c" (eng. to become greener). This case is particularly important as verb having is often used for detecting facts about attributes and requires special handling not to loose information.

VII. CONCLUSIONS AND FURTHER WORKS

As can be seen from the results of our evaluation automatic fact extraction from texts in Polish language is still very difficult, especially when using simple sentence analysis based solutions. The vast majority of errors still occur at tagging level done by external (evaluated) tools. Especially the dependency parser still generates a lot of errors. It has to be noted that the quality of the processing by the Multiservice is gradually getting better as the tool is in constant development.

The correct morpho-syntactic analysis of Polish language is now a major challenge for the field of machine processing of the Polish language. Further research should be done in this field. Unless then solutions like the TEAMS software can be used only for preliminary facts extraction requiring further verification. This can be however done using crowdsourcing mechanisms. Such work is already being done [40]. Even at current state of morpho-syntactic analysis, combining TEAMS results with crowdsourcing based verification can produce a viable results. For on the fly knowledge base creation it is however to early and further research needs to be done.

REFERENCES

[1] C. D. Manning, P. Raghavan, H. Schütze *et al.*, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008, vol. 1, no. 1.

[2] M. Bates, "The theory and practice of augmented transition network grammars," in *Natural language communication with computers*. Springer, 1978, pp. 191–254.

[3] C. D. Manning, H. Schütze *et al.*, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.

[4] F. Zanettin, *Translation-driven corpora: Corpus resources for descriptive and applied translation studies*. Routledge, 2014.

[5] J. Szymański, "Words context analysis for improvement of information retrieval," in *International Conference on Computational Collective Intelligence*. Springer, 2012, pp. 318–325.

[6] W. Duch, J. Szymański, and T. Sarnatowicz, "Concept description vectors and the 20 question game," in *Intelligent Information Processing and Web Mining*. Springer, 2005, pp. 41–50.

[7] J. Szymański, "Self–organizing map representation for clustering wikipedia search results," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2011, pp. 140–149.

[8] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5, 2010, p. 3.

[9] A. Gangemi, V. Presutti, D. Reforgiato Recupero, A. G. Nuzzolese, F. Draicchio, and M. Mongiovì, "Semantic web machine reading with fred," *Semantic Web*, no. Preprint, pp. 1–21, 2016.

[10] A. Gangemi, V. Presutti, F. Draicchio, A. Musetti, and A. Nuzzolese, "TIPALO," http://wit.istc.cnr.it/stlab-tools/tipalo, 2012, [Online: 10.05.2017].

[11] M. Bańko, "Word order in sentence, (in Polish)," http://sjp.pwn.pl/poradnia/haslo/szyk-wyrazow-w-zdaniu;14378.html, 2013, [Online: 10.05.2017].

[12] ——, "Adjective order in sentence, (in Polish)," http://sjp.pwn.pl/poradnia/haslo/szyk-przymiotnikow;12630.html, 2013, [Online: 10.05.2017].

[13] D. Feng, *Factorizing information extraction from text corpora*. ProQuest, 2007.

[14] A. Savary, M. Ogrodniczuk, M. Zawislawska, K. Glowinska, and M. Kopec, *Coreference: Annotation, Resolution and Evaluation in Polish*. Walter de Gruyter GmbH & Co KG, 2015.

[15] D. A. Ferrucci, "Ibm's watson/deepqa," in *ACM SIGARCH Computer Architecture News*, vol. 39, no. 3. ACM, 2011.

[16] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller, "Watson: beyond jeopardy!" *Artificial Intelligence*, vol. 199, pp. 93–105, 2013.

[17] M. Woliński, "Komputerowa weryfikacja gramatyki świdzińskiego," Ph.D. dissertation, Instytut Podstaw Informatyki PAN, Warszawa, 2004.

[18] M. Świdziński, *Gramatyka formalna języka polskiego*. Wydawn. Uniwersytetu Warszawskiego, 1992, no. 349.

[19] M. Piasecki, "Polish tagger takipi: Rule based construction and optimisation," *Task Quarterly*, vol. 11, no. 1-2, pp. 151–167, 2007.

[20] Z. Saloni, W. Gruszczyński, M. Woliński, R. Wołosz, and D. Skowrońska, "Morfeusz, (in Polish)," http://sgjp.pl/morfeusz/index.html, 2013, [Online: 10.05.2017].

[21] D. Czerski, K. Ciesielski, M. Dramiński, M. Kłopotek, P. Łoziński, and S. Wierzchoń, "What nekst?âĂŤsemantic search engine for polish internet," in *Challenging Problems and Solutions in Intelligent Systems*. Springer, 2016, pp. 335–347.

[22] M. Ogrodniczuk and M. Lenart, "Web service integration platform for polish linguistic resources." in *LREC*, 2012, pp. 1164–1168.

[23] J. Waszczuk, "Harnessing the crf complexity with domain-specific constraints. the case of morphosyntactic tagging of a highly inflected language." in *COLING*, 2012, pp. 2789–2804.

[24] A. Buczyński and A. Przepiórkowski, "Spejd: A shallow processing and morphological disambiguation tool," in *Language and Technology Conference*. Springer, 2007, pp. 131–141.

[25] J. Waszczuk, K. Głowińska, A. Savary, A. Przepiórkowski, and M. Lenart, "Annotation tools for syntax and named entities in the national corpus of polish," *International Journal of Data Mining, Modelling and Management*, vol. 5, no. 2, pp. 103–122, 2013.

[26] M. Kopec and J. Kazimierza, "Zero subject detection for polish." in *EACL*, 2014, pp. 221–225.

[27] A. Wróblewska and A. Przepiórkowski, "Projection-based annotation of a Polish dependency treebank," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Reykjavík, Iceland: ELRA, 2014, pp. 2306–2312. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/index.html

[28] E. Rudnicka, W. Witkowski, M. Kaliński *et al.*, "Towards the methodology for extending princeton wordnet," *Cognitive Studies| Études cognitives*, no. 15, pp. 335–351, 2015.

[29] Wikipedia, "Polyp, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Polip_(biologia)}

[30] ——, "Cnidoblasts, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Knidoblast}

[31] ——, "Box jellyfish, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Kostkowce}

[32] ——, "Medusa, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Meduza}

[33] ——, "Anemones, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Ukwia%C5%82y}

[34] ——, "Wojciech Jaruzelski, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Wojciech_Jaruzelski}

[35] ——, "Lech Wałęsa, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Lech_Wa%C5%82%C4%99sa}

[36] ——, "Aleksander Kwaśniewski, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Aleksander_Kwa%C5%9Bniewski}

[37] ——, "Lech Kaczyński, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Lech_Kaczy%C5%84ski}

[38] ——, "Bronisław Komorowski, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Bronis%C5%82aw_Komorowski}

[39] ——, "Andrzej Duda, (in Polish)," 2017, [Online: 10.05.2017]. [Online]. Available: \url{https://pl.wikipedia.org/wiki/Andrzej_Duda}

[40] T. Boiński, "Game with a purpose for mappings verification," in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE, 2016, pp. 405–409.