Linguistic engineering tools

I Don't Like Munday

MACIEJ OGRODNICZUK
Institute of Computer Science
Polish Academy of Sciences, Warsaw
maciej.ogrodniczuk@ipipan.waw.pl
Maciej Ogrodniczuk earned his master's degree at the Faculty of Mathematics, Informatics and Mechanics at the University of Warsaw, and he holds a doctorate in the humanities. He works on the Linguistic Engineering Team of the Department of Artificial Intelligence at the PAS Institute of Computer Science. He leads the project "Machine methods of identifying references in Polish texts," running between 2011 and 2014.

Machine linguistic tools are now so widespread that we barely notice them. But we don't like to conform to their whims – rather, we want them to adapt to us, to start to understand what we say or write, be it in English, Polish, or otherwise. This requires effective text analysis methods, and digital tools are becoming increasingly sophisticated in response

Picking up a phone and dictating emails or text messages is no longer the realm of science fiction. Automatic translation - perhaps still a little clunky, but already allowing us to gain a basic understanding of a text written in a foreign language? Sure! How about speech synthesis? We often don't even realize that announcements made at airports or stations are computer-generated. When we combine these tools and add computational power - most likely taken from the cloud (infrastructure providing it remotely and on demand) - we can imagine having a sufficiently comprehensible conversation with a Chinese or Thai speaker, with a smartphone acting as an intermediary. All this is made possible by linguistic engineering tools, which are also becoming increasingly available for Polish. Simple? Only in theory, since the entire process relies on solutions to many complex problems.

I'm Hungary...

Once our digital assistant gets to grips with converting speech signals into phonemes and attempts to put them together to form text, they almost always end up with numerous variants. Did the speaker say "rain," "rein" or "reign"? "Anteater," "aunt eater" or perhaps "and eat her"? Countless such choices need to be worked out. And if we want automatic translation, then that's just the beginning: content needs to be analyzed in greater depth to detect proper nouns ("hungry" or "Hungary"? "turkey" or "Turkey"?), resolve syntactic relations within sentences, and deal with any other linguistic,

Illustrations by Dawid Ryski

orthographic and grammar quirks specific to the given language (such as free and often discontinuous word order in Polish). Things become even more complicated when a more advanced way of converting text is needed, for example to automatically generate a summary. We need to take into account complex relationships between parts of the utterance, such as coreference (more than one reference to the same object or concept), to ensure the result is stylistically consistent, for example by replacing a pronoun in the summary with its full equivalent. Synthesis of speech from a text analyzed this way is now relatively simple; in fact one of the most effective solutions currently on the market was developed by a Polish company.

It is clear that what lies at the root of these complexities is the multifaceted ambiguity of language. In our daily interactions, we resolve this contextually using all the means at our disposal, such as the likelihood of the given construction ("I don't like Monday" – referring to the day, being generally more likely than "I don't like Munday" – referring to a town in Texas), general world knowledge, non-verbal signals, and very well-developed abilities to read and anticipate speaker intentions. The task is far more difficult for computers, albeit it is slowly becoming possible.

No more "should of"!

We have already made significant headway. We can cope with such multilayered ambiguity through in-depth analysis of the source text on different levels: trying to properly divide it into words and sentences, disambiguating

grammatical categories, selecting the right meaning of words, identifying the grammatical structures, and even conducting deeper semantic analysis.

Let's start with text segmentation and grammar-checking – basic tasks that bring enormous benefits. For both English and Polish, there exist professional software packages that automatically verify the prescriptive "correctness" and stylistic appropriateness of text with capabilities far exceeding those of popular desktop editing packages. They detect forms of words or phrases that are incorrect despite being in widespread use (English examples include "should of," "could of"), repetitions in a context broader than just simple duplication ("the whole class behaved with great class"), words that share a common core ("she misplaced the cookbook in a different place"), inconsistencies in spellings of proper nouns ("Hoffman" vs. "Hoffmann" appearing in the same document), punctuation errors (lower case following a full stop at the end of a sentence, a comma before opening a bracket), and so on. Such software also provides hints as to where mistakes might be potentially lurking ("in consistent" rather than "inconsistent", "fort he" rather than "for the"), and may flag frequently confused homophones ("their" vs. "there" or "its" vs. "it's"). For Polish, such software can additionally double-check the grammatical inflection of words.

As for syntactic analysis, considerable progress has been made for English and is also quite advanced for Polish. Tools include component, dependency and functional analysis, machine implementations of various grammars, and valency dictionaries (specifying what kind of complements specific verbs are usually linked to). Semantic relationships (such as synonyms, antonyms, hyponyms, etc.) are successfully modeled in systems such as WordNet – a database of relationships between lexical units – and its Polish equivalent, called Słowosieć. Here are creating ontologies and other models of general knowledge, once again to be harnessed as a linguistic tool.

At the same time, systems are being developed that work on all these linguistic levels simultaneously.

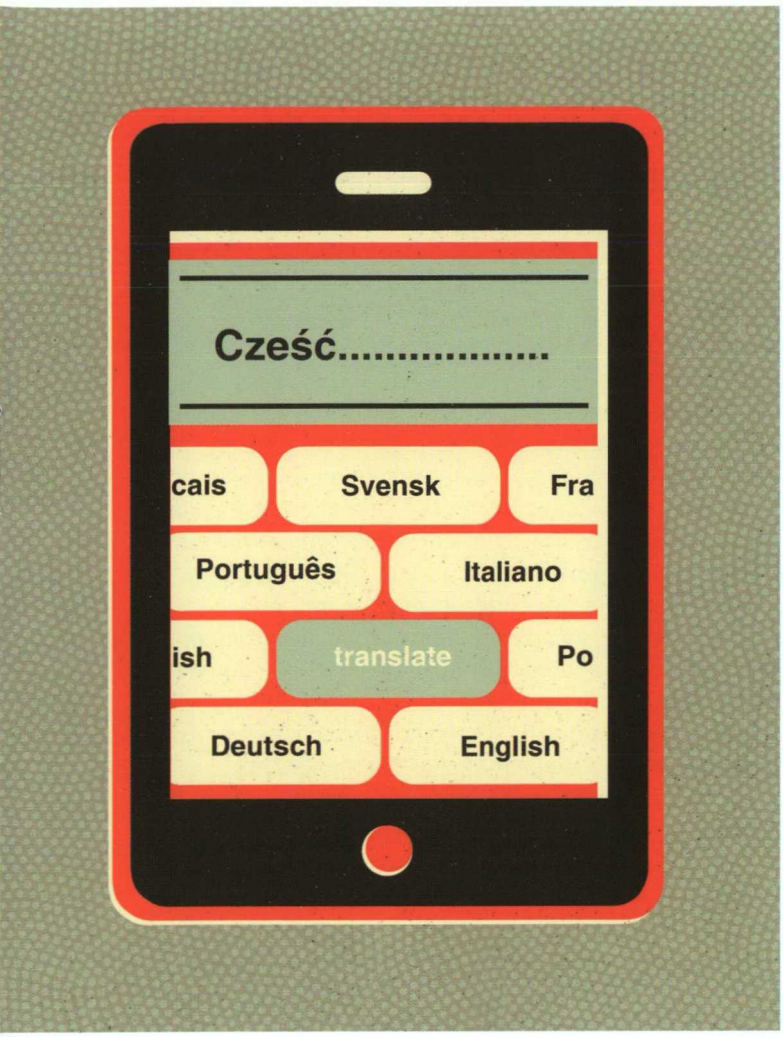
They are used, for instance, to evaluate the readability of texts, which can help facilitate interactions between administrative institutions and the people they serve, improve the accessibility of technical manuals and textbooks, and so on. As well as traditional methods based on word and sentence length, they also take account of lexical (such as the obscurity or polysemy of individual words), syntactic and morphosyntactic factors (the presence of participles, negations, and sentences containing numerous clauses). Advancements are also being made in combining multilingual tools, such as models that utilize automatic translation of texts, process them with more sophisticated tools for the target language, and then convert the properties so discovered back to the source language.

Linguistic tools frequently form a part of larger systems (e.g. content management systems), as well as providing an option to classify texts automatically, generate lists of documents similar in content, and extract key words and phrases. Online auction sites are already able to automatically translate descriptions of goods on offer into a user's language of choice, allowing them to extend their potential buyer base. There are also growing numbers of web services allowing users to obtain answers to questions posed in natural language, such as "When was the Eiffel tower finished?", once again thanks to linguistic engineering harnessed for the analysis of web content.

Structure vs. statistics

For many years, theoretical and empirical approaches competed in language analysis (although now they are increasingly being used concurrently). The aim of the former is to analyze language in terms of its abstract structure, while the latter focuses on the effectiveness of the processing of authentic linguistic data stored in corpora, with all the consequences this brings, taking into account real complexity, errors, and so on. The theoretical approach is based on linguistic analysis using rule-based methods and tools created on the basis of an idealized model of language. The domain of the empirical approach is broadly-construed statistical analysis of actual linguistic phenomena.

Critics of the empirical approach point out the need to model full linguistic competence; this is not reflected in any collections of examples, regardless of their size, a direct result of language's limitless potency. Corpus linguists respond to these arguments by providing difficult examples of actual data not covered by even the most sophisticated theories. The two approaches have been di-



verging and converging since the 1950s; after recent years, which have been a golden era for the statistical approach, it seems that we are now observing a return to linguistic methods, as shown by the recent extension of statistics-based machine translation systems to include grammatical rules, vastly improving their effectiveness. One good example of the two approaches intertwining can be found in a new method of automatic learning of linguistic (morphological) information from unprocessed data, which suggests that the two methods will largely continue to complement each other.

When will a computer say "I'm sorry"?

The significance of linguistic resources and tools continues to increase, in part due to the exponential growth of information; Facebook and Twitter alone generate hundreds of terabytes of data every day. Google will never replace linguists not only because current search engines do not provide advanced linguistic description of content, but also because of the inability of online models to fully represent any given language. We continue to need reference collections of texts such as corpora, as well as linguistic tools including dictionaries, banks of concrete syntax trees, and databases of semantic relationships, all of which are increasingly being developed on the basis of corpora.

In the longer term, the future of linguistic engineering lies in computers being gradually equipped with the ability to understand the semantics of utterances, draw conclusions and aggregate information, for example in order to create concise summaries of large numbers of documents. For HAL 9000, the intelligent computer from Arthur C. Clarke's "2001: A Space Odyssey," to be able to inform an astronaut that it refuses to let him back into the spaceship with the sentence "I'm sorry, Dave, I'm afraid I can't do that," it would need to be able to understand humans, interpret their words, relate them to reality, make a decision (in this case rejecting an order in spite of being able to carry it out), formulate a response (choosing the right words, in this case politely refusing the request), and transmit it in an understandable way. Even this seemingly simple action goes far beyond simply processing words and sentences, requiring instead a general artificial intelligence, able to interpret intentions and make judgments, and perhaps even having a certain level of self-awareness. Such systems are likely a long way off; however, automated assistants that are reasonably competent in analyzing information and executing commands given using natural language do seem to be just around the corner. ■

Further reading:

CLIP - Computational Linguistics in Poland. http://clip.ipipan.waw.pl

Jurafsky D., Martin J.H. (2008). *Speech and Language Processing* (2nd ed.). Prentice Hall Series in Artificial Intelligence, Prentice Hall.

Przepiórkowski A., Bańko M., Górski R.L, Lewandowska-Tomaszczyk B. (eds.) (2012). *Narodowy Korpus Języka Polskiego* [National Corpus of Polish]. Warsaw: PWN. (full set of papers in Polish available online: http://nkjp.pl/settings/papers/NKJP_ksiazka.pdf.).

Wilks Y. (2005). The History of Natural Language Processing and Machine Translation. *Encyclopedia of Language and Linguistics*.