The Importance of Algorithms in Language Analysis

Languages are how we communicate. They allow the thoughts of one person to be expressed to another in sophisticated ways, but they are not exclusive to humans. Machines and computers have their own language to communicate with other machines. However, there is a large gap between the understandings of machine and human language. While humans can, through some effort, learn machine language and understand what the machine is trying to output and why, it is extraordinarily difficult for machines to understand human language. This is made even more difficult due to the numerous languages and dialects in use and the way those languages are spoken.

The core of this difficulty stems from the human ability to understand words through context or context clues. For example, the phrase “The trophy will not fit in the suitcase, it is too small,” has ambiguous wording in it that a normal person might not have noticed because it was so trivial. The word “it” is ambiguous in this sentence. A normal person would automatically assume that the suitcase is too small, as referring to the trophy being too small to fit would not make sense. However, a machine has no way to use human intuition to figure out what “it” is referring to. This creates a large problem for artificial intelligence, as it would be very difficult or impossible for a machine to understand human speech or writing.

To solve this issue, numerous algorithms have been created and devised to break human language down to its barest elements and remove naturally present ambiguity. This is known as Word Sense Disambiguation or WSD, a problem that has plagued the field of artificial intelligence for many years. Due to the difficulty of this problem, WSD is also known as an AI complete problem, a problem that is so difficult, it is equivalent with solving the central issue of artificial intelligence – making AI as smart as a human. This would allow a machine to understand human language with much greater accuracy than previously before. For the most part, WSD is nearly complete for English due AI being developed mostly in America during its early stages. However, many languages such as Hindi, Nepali and Arabic are still in development due to the complexity of those languages and the amount of time invested in them.

The typical process for Word Sense Disambiguation is as follows: Tokenization, Part-of-Speech Tagging, Lemmatization, Chunking and Parsing. Tokenization is dividing the text into basic units or tokens called words. Part-of-Speech Tagging is determining the grammatical category for each word that is found. Lemmatization refers to the morphological analysis of the words to find the root words, the ambiguous words. From there the text is partitioned into syntactically correlated parts, chunking, and then parsed - providing the parse tree of the sentence structure.

There are 2 widely accepted solutions for WSD. There are supervised methods, in which there is training through data from resources like dictionaries, thesauri and WordNet, a lexical database for the English language. This method is relatively accurate but takes a lot of time to train. The other method is an unsupervised method in which training is skipped, but accuracy is generally too low for reliable use.

However, a new method has appeared in the form of unsupervised graph-based techniques. These methods don’t require extensive training, yet have boosted their accuracy higher than the previous unsupervised methods. This method works by using constructed semantic graphs for words to be disambiguated by taking into account nodes from thesauri like WordNet. Then node ranking or node activation algorithms are used to determine the best candidate sense for each word.

There are 6 popular algorithms to measure the word semantic similarity: the Leacock & Chodorow, the Lesk, the Wu-Palmer, the Resnik, the Lin, and Jiang & Conrath algorithms. These algorithms determine the dependency between word sense represented as nodes in the graph. Then 4 graph based algorithms determine the closeness of the nodes and the nodes with the highest closeness is assigned the sense for that particular word.

Of the 6 popular algorithms, the Lesk algorithm is generally the most popular. The Lesk algorithm works to find the correct meaning of each word in context by individually locating the sense that overlaps the most between its dictionary definition and the given context. Rather than simultaneously determining the meanings of all words in a given sentence or string, this approach takes each word individually, independent of the meaning of the other words occurring in the same context. While the Lesk algorithm is typically very accurate, it is very sensitive to the exact wording of the particular definition that is used and only uses the glosses of senses of the word being considered. This is a pitfall because the dictionary may not have a complete set of glosses or may not have an extensive enough set for the algorithm to use, severely limiting the accuracy of the algorithm.

Since the Lesk algorithm is generally the most popular algorithm for WSD, it has been used in WSD implementations for languages other than English as well. A prevalent language where the Lesk algorithm and modified versions of it are used is the Arabic language. While the Lesk algorithm has been found to be somewhat accurate in Arabic, modified versions of the algorithm using measures of the methods above (Leacock & Chodorow, Resnik, Wu-Palmer, Jiang & Conrath) have also been used to disambiguate words. These variations have led to more accurate translations. While the unmodified Lesk algorithm achieved an average accuracy of around 59%, the modified algorithm under the Leacock& Chodorow similarity measure has achieved an average accuracy of around 67%. Further modifications and advances will eventually lead to the level of translation success for Arabic as it is for English. Wherever the path of linguistic translation leads, the Lesk algorithm will be a popular tool to use.

The Lesk algorithm and Word Sense Disambiguation are extraordinarily important to the field of artificial intelligence. This hurdle must be overcome for the term to really have any meaning. There is little enthusiasm for artificial intelligence that only scientists can communicate with, and without that interest, there is no funding for such projects. But with steps to helping machines understand the complexities and nuances of human speech, including dialects other than English, the goal of reaching full-fledged artificial intelligence inches one step closer. A pervasive area of its use is in data search engines like Google, Yahoo, and Bing. To make results more accurate and meaningful to the user, WSD must be used in order to avoid misunderstanding what the user is really trying to ask. However, the most important and obvious application for WSD and the Lesk algorithm would be for translation between different languages. It is important for the translator not to get confused by the ambiguous words in order to get an accurate translation. This has a huge impact on society because it is far easier to understand each other’s language than what was previously possible. To understand what another says in a different language, it is no longer required to be fluent in that language or to have someone translate either. This makes it possible for business in different countries to communicate with each other more effectively and for lower cost, allowing for easier cooperation between different entities, business or otherwise. Word Sense Disambiguation will be crucial not only to artificial intelligence in the future, but also to the future of humanity and human cooperation and will only advance in complexity and accuracy in the years to come.

Citation:

Srinivas, Mulkalapalli, and B. Padmaja Ran. "Word Sense Disambiguation Techniques for Indian and Other Asian Languages: A Survey." *International Journal of Computer Applications* 156.8 (2016): 35-41. *ProQuest*. Dec. 2016. Web. 7 July 2017. <http://search.proquest.com/docview/1913919486?pq-origsite=gscholar>.