Natural Language Processing Tasks

Natural language processing is a field of computer science that focuses on the interactions between human language and computers. It involves the development of computational methods to analyze, understand, and derive meaning from human language. By using natural language processing, developers can organize and structure knowledge to perform tasks, such as sentence boundary detection, part of speech tagging, named entity recognition, and relationship extraction. Natural language processing tasks can be separated into two types, low-level task and high level tasks. It is important to note that some of these tasks have direct applications, while others are sub-tasks that are used to help solve larger tasks. It is also standard practice to think of some levels of analysis as feeding into others, where typically, low-level tasks, feed into high-level tasks. Low-level natural language processing tasks include sentence boundary detection, tokenization, part-of-speech tagging, morphological decomposition, shallow parsing, and problem-specific segmentation. Let's have a look at these in more detail. Sentence boundary detection is a problem of deciding where sentences begin and end. It is a critical first processing step for many natural language processing applications. In general, natural language processing tools require the input to be divided into sentences. However, identifying sentence boundaries can be challenging. One reason for this is because punctuation marks are often ambiguous. In the context of a pathology report, items in a list, abbreviations and titles such as doctor, complicate this task. Tokenization recognizes individual tokens, such as word or punctuation in a sentence. This is an important task, as it identifies units that do not need to be further decomposed for subsequent processing. Errors made at this stage are very likely to produce more errors at later stages of processing. This therefore will have a huge impact on the end result. Within a pathology report, tokens often contain characters, typically used as token boundaries, for instance, hyphens and forward slashes. Hyphens are common in drug names, such as 2-acetoxybenzoic acid, which is a chemical name for aspirin. Forward slashes, are regularly found in recommended drug dosages, such as 10mg/day. Part-of-speech tagging is also known as part-of-speech assignment to individual words. It is the process of marking up a word in a text that's corresponding to a particular part-of-speech, based on both its definition and its context. For example, its relationship, with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is when we identify words as nouns, verbs, or adjectives. However, verbs ending in 'ing' that are used as nouns, complicate this task. Part-of-speech tagging is an important task as it can provide a lot of information about a word and the words near it. For example, adjectives are often followed by nouns. Therefore, a useful feature of part-of-speech tagging is finding specific words or phrases, such as people or organizations in text. The pronunciation of a word also depends on part-of-speech tagging, as it will break words down into syllables. Moving on to morphological decomposition. Morphology is the study of the structure of words. A morpheme is the smallest meaningful unit of language, hence, morphological decomposition is the process of establishing the morphemes, from which a given word is constructed. It's an important task as an application must first recognize the word in question, before analyzing it at any level. For example, the word boxes can be decomposed into box, which is the root or stem of the word. And es, which is a suffix indicating the plural form of the word. Many medical terms need to be decomposed in order to understand them. For example, the word nasogastric, where naso is referring to the nose and gastric is referring to the stomach. Therefore, a nasogastric tube is a tube that is passed through the nose to reach the stomach. Morphological decomposition is also essential in applications ranging from spell-checker to machine translation. Shallow parsing is another low-level natural language processing task, also known as chunking. It is the process of sentence analysis, which first identifies the main components of sentences, such as nouns, verbs, or adjectives, and then links them to higher-order units that have discrete grammatical meanings, such as noun groups or phrases. Finally, problem-specific segmentation is the process of dividing text into meaningful units, such as words, sentences, topics, or sections. This could be the past medical history section within a personal medical record. Now let's have a look at some high-level natural language processing tasks. High-level tasks build on low-level tasks, and are usually problem-specific. High-level tasks include spelling or grammatical error identification and recovery. Named entity recognition, word sense disambiguation, negation, and uncertainty identification, relationship extraction, temporal interferences, and information extraction. Let's start with spelling or grammatical error identification and recovery. This task is mostly interactive as it is far from perfect. As you can imagine, many different errors can occur. This could be when correct words are flagged as errors, leading to what we call, false positives, or when identically sounding, differently spelled words are used incorrectly, leading to false negatives. Named entity recognition is the task of finding and classifying names in text. More specifically, it is the process of identifying specific words or phrases, also known as entities, and arranging them into predefined categories, such as persons, locations, diseases, genes, or medication. Named entities identified in text can then be indexed or linked off. This task is widely used when searching for specific information within pathology reports. Moving on to word sense disambiguation. This task assigns the appropriate meaning or sense to a given word in a text. An example of this is the word bear, B-E-A-R, which could be the verb, meaning to carry or support, or it could be the noun, meaning the animal. This word is identical in spelling and pronunciation, but differs in meaning and grammatical function. Negation and uncertainty identification is a task in which it infers, whether a named entity is present or absent, and quantifies uncertainty of that inference. Negation can be explicit. For example, patient denies chest pain. Or implied, for example, lungs are clear upon auscultation, which implies the absence of abnormal lung sounds. Negated or affirmed concepts can also be expressed with uncertainty, as in the ill-defined density suggests pneumonia. These are just a few simple examples of negation and uncertainty identification in a medical context. Relationship extraction is the process of determining relationships between entities or events. For example, who is married to whom? In a clinical report, this would mean words like treats, causes, and occurs with. Moving on to temporal inferences, this is the process of making inferences from temporal expression and temporal relations. For instance, inferring that something has occurred in the past, or may occur in the future. An example of this would be, symptoms began after medication x was administered. Information extraction is the task of finding and understanding limited relevant parts of text, where it collects information from many pieces of text, and provides the relevant information in a structured form. Therefore, information extraction enables information to be organized, so that it is useful to people. The previously mentioned high-level tasks are often part of a larger information extraction task. It is worthy to note that any practical natural language processing task includes several sub-tasks. For example, low-level tasks are carried out sequentially before high-level tasks. In addition, a mix and match of different tasks is possible. This can be thought of as a pipeline. As different algorithms may be used for a given task, then the output of one analytical task becomes the input to the next task. Why should we use pipelines in natural language processing? If it's a complicated task, then a pipeline can help break it down into sub-tasks that can be solved independently. This also means that each sub-task can be accomplished using different algorithms, and you may want to swap the algorithms for another one, without affecting the rest of the pipeline. Now that we have covered the main tasks in natural language processing, in the next video, we will talk about the computational methods used in natural language processing.