ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/291179558

Natural Language Processing

Chapter · January 2013

DOI: 10.1007/978-1-4419-9863-7_158

CITATIONS
9

READS
13,605

2 authors:

Karin Maria Verspoor
University of Melbourne
328 PUBLICATIONS **5,798** CITATIONS
SEE PROFILE

Kevin Bretonnel Cohen
University of Colorado
223 PUBLICATIONS **5,926** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

BioNLP View project

ExpoInfo View project

All content following this page was uploaded by Karin Maria Verspoor on 07 June 2018.

The user has requested enhancement of the downloaded file.

Title: Natural Language Processing

Name: Karin Verspoor, Kevin Bretonnel Cohen

Affil./Addr. 1: University of Colorado Denver

PO Box 6511, MS 8303, Aurora, CO 80045

karin.verspoor@ucdenver.edu

Affil./Addr. 2: kevin.cohen@gmail.com

Natural Language Processing

Synonyms

Computational Linguistics

Text Mining

Text Processing

Natural Language Understanding

Information Extraction

Biomedical Natural Language Processing (BioNLP)

Definition

Natural Language Processing is the analysis of linguistic data, most commonly in the form of textual data such as documents or publications, using computational methods. The goal of natural language processing is generally to build a representation of the text that adds structure to the unstructured natural language, by taking advantage of insights from linguistics. This structure can be *syntactic* in nature – capturing the grammatical relationships among constituents of the text – or more *semantic* – capturing the meaning conveyed by the text.

Natural Language Processing is used in systems biology to develop applications that integrate information extracted from the literature with other sources of biological data (see Applied Text Mining).

Characteristics

The typical natural language processing system consists of a pipeline of components that manipulate an input text in increasingly sophisticated ways. Generally, the aim of each component is to add structure to the text that can be used to facilitate downstream processing. The components early on in the pipeline handle tasks that are close to the surface strings of the text, while later components aim to analyze concepts and relationships. Various methods may be used to accomplish component tasks, ranging from rule-based methods such as regular expressions and finite state automata, to statistical and machine learning models.

In Figure 1, we can see an example of the processing of a single sentence from a biomedical text. Each level will be discussed in more detail below.

Tokenization and Sentence Demarcation

Natural language processing is strongly word-based, in that words are generally considered to carry the meaning of a text. It is therefore important as a pre-processing step to any further analysis to delimit the individual *word tokens* that make up a text. This is seen at the "Word" level in Figure 1. This process is referred to as *tokenization*. While a simple approach is to split the text on any whitespace or punctuation, some care must be taken in biomedical texts to appropriately handle punctuation that has special meaning in certain contexts, such as a single quote in the representation of a DNA strand (*5'-GCRTGNCCAT-3'*), the characters in some chemical names (*tricyclo(3.3.1.13,7)decanone*), hyphens which can indicate charge (*Cl-*), constitute part

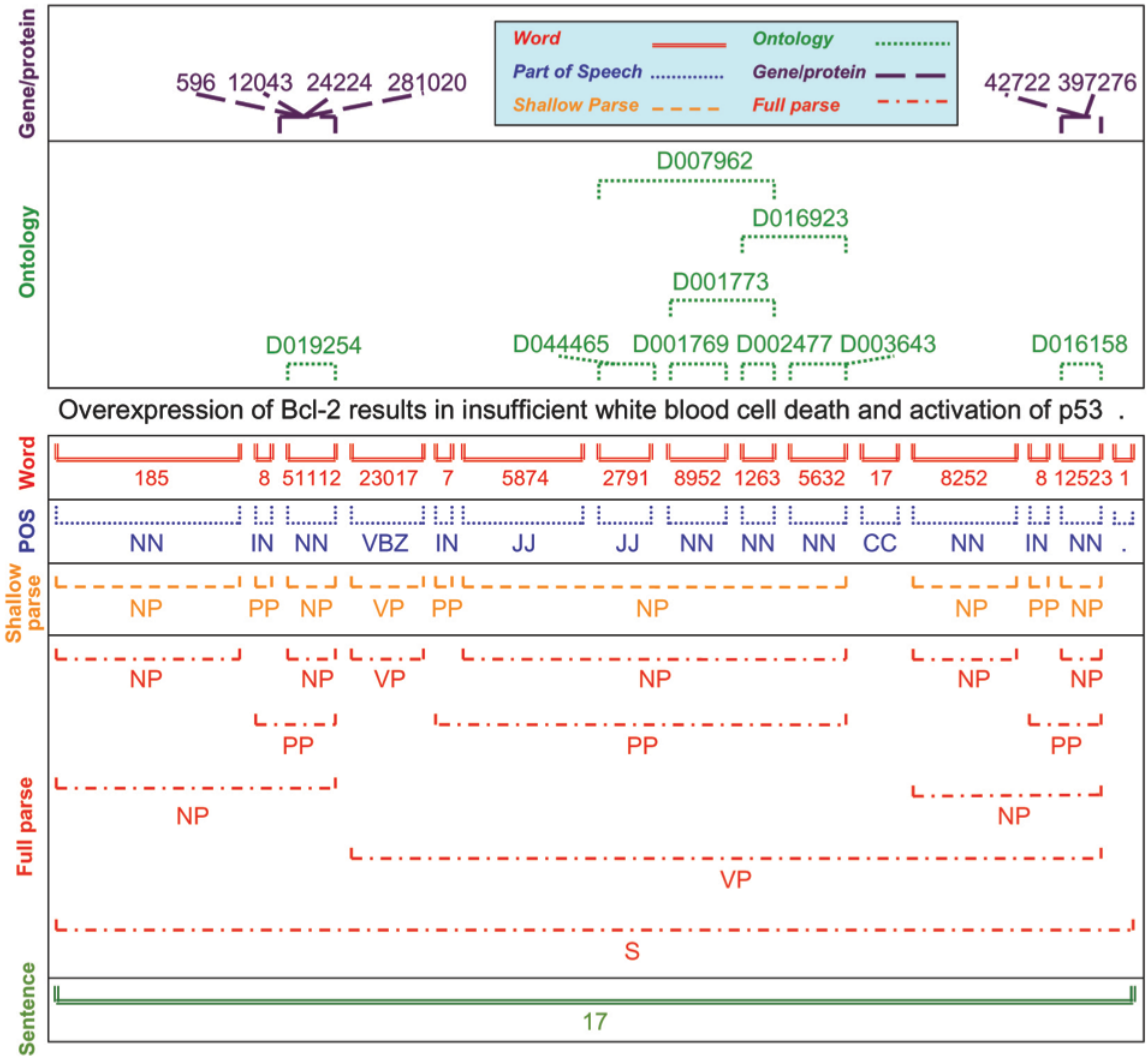

Fig. 1. The levels of analysis for a sentence processed by a typical NLP pipeline. From Hunter and Cohen, 2006 (6), adapted from Nakov et al, 2005 (10).

of a gene or cell name (*hsp-60*, *t-cell*), or a knocked out gene (*lush- flies*), etc. Thus tokenization tools sensitive to the biomedical context are required.

As a precursor to syntactic analysis, it is also important to delimit individual sentences within a text. This is because the sentence is generally the grammatical unit of a text. Similarly to tokenization, *sentence splitting* generally involves taking advantage of a basic heuristic: look for normal sentence-final punctuation (period or question mark) followed by a capital letter. However, again complications can be introduced in the context of names containing initials (e.g. *H.G. Wells* or *Dr. Bronner's soap*) where the heuristic would incorrectly split a sentence into multiple pieces. Similarly,

domain-specific conventions that sometimes require a sentence-initial lower-case letter can cause problems for sentence demarcation, such as

> The process of activation involves [...] phosphorylation of tyrosine **kinases. p21(ras)**, a guanine nucleotide binding factor, mediates T-cell signal transduction ... (from PMID 8887687, with thanks to Bob Carpenter for finding it)

Syntactic and Morphological Analysis

Syntactic information about the text can be important to assist in resolving ambiguities and in establishing the appropriate relations among the words in a text. At the most basic level, determining whether a word is a noun or a verb (or some other part of speech) can be useful. This is accomplished through tools that perform *part of speech (POS) tagging*. Then, identification of phrases in the text can be important, such as recognizing that a sequence of words forms a single conceptual unit (e.g. *breast cancer*, *NF kappa beta inhibitor*). A commonly used strategy for this is *shallow parsing*, which involves identifying coarse phrasal structures, such as noun phrases, without identifying the specific grammatical relationships among them. In contrast, *Deep parsing* determines the full set of grammatical relations among words in a sentence, producing a complete *parse tree* to represent these relations.

The surface forms of words will vary depending on their syntactic usage in a sentence, for instance a noun appearing in plural form or a verb appearing in various tenses (*regulated, regulating, regulates*). Often, it is desirable to normalize such variation to a base form of the word in order to appropriately associate different occurrences of the same term. This is called *morphological normalization* and is often accomplished in practical NLP applications through *stemming* tools which strip off inflected word endings. The *Porter* algorithm, based on suffix stripping, is a popularly used strategy for stemming (11).

Information Extraction

Information extraction in general refers to the extraction of specific types of information from text, and normally formalized in a structured representation, such as an event template or a concept from an externally-defined ontology. It can refer to the association of particular strings of a text to a category of interest, for instance identifying protein names in a publication.

Named Entity Recognition

In the upper levels of Figure 1 we see annotations of ontology terms and gene/protein terms. Many of such terms correspond to *named entities*, that is, to objects that are generally referred to by name. This is in contrast to terms that correspond to processes or events, which normally require identification of higher-order relations. Examples of named entities in the biological domain that are often targeted for extraction are genes, diseases, chemicals, or experimental methods.

Various methods exist for performing named entity recognition. The most basic approach is to compile a dictionary of the relevant names for a specific category of entities, and to perform a string match into the dictionary. Empirical methods based on supervised machine learning will often use a dictionary match as one feature of a model that also considers surrounding words, syntax, and other textual evidence to identify likely instances of terms from a particular category.

Relation and Event Extraction

Beyond extraction of entities, many applications require extraction of *relations* among those entities. One popular example, addressed in several shared tasks such as BioCreative (5; 8; 9) and BioNLP'09 (7), is identification of protein-protein interactions from text. This first requires the recognition of the proteins as entities, and then identi-

fication of an interaction relation among at least two of the recognized proteins. In the sentence in Figure 1, for instance, we can identify an activation relationship between Bcl-2 and p53, i.e. one of the key pieces of information in the sentence can be summarized as *Bcl-2 activates p53*. Strategies for relation extraction again vary from high-precision linguistic-based methods (1) to high-recall supervised learning methods (2), and hybrid methods that achieve more balanced performance (4).

Co-reference resolution

Co-reference resolution refers to identifying multiple occurrences in the text of the same entity or event. It includes resolving pronouns such as "it" to their references, as well as other kinds of references such as definite noun phrases (a noun phrase that starts with "the", e.g. "the protein"). Note that these references can include references to events previously mentioned, e.g. "the process" or "this interaction".

Implementation Aspects

Natural Language Processing systems are implemented in the form of software. Such systems tend to have modular architectures where components such as those outlined above are run serially in a "pipeline".

Document format issues

Before any more sophisticated linguistic processing can be performed, documents must be converted in a form that is easy for computational tools to work with. Since source documents can be available in various formats, including HTML, XML, Microsoft Word, and PDF in addition to plain text, NLP systems must clearly specify the kinds of input documents they can handle. In general documents must be converted to a simpler plain text representation without the structure and formatting information

available in other formats. There are tools available to assist with these conversions, but they can vary in quality and effectiveness.

In addition, NLP systems must be sensitive to the character encoding of a given document. Documents can be encoded in numerous formats, including UTF-8 and ISO-8859-1. Some characters, in particular special two-byte UNICODE characters such as Greek letters, will not be correctly interpreted if the correct encoding is not utilized when loading the document. Since such characters can be meaningful in biomedical texts, (e.g. in the name of the TGF-β gene), this is an issue that cannot be overlooked.

For most applications, it is preferable to retain as much of the original document structure as possible. Certain formatting information can have semantic import. For instance, italics are sometimes used to highlight a gene name in a document. In addition, sensitivity to the sections of a document can provide a system with an advantage in solving certain problems, such as for detecting *new* experimental protein interactions described in the text – one would not expect these to be mentioned in a background or methods section. Document sections are generally most reliably identified by taking advantage of the previously demarcated structure of the document, but sophisticated algorithms to perform document zoning might need to be employed if such demarcations are unavailable.

Unstructured Information Management Architecture

The Unstructured Information Management Architecture, or UIMA, is a commonly used architecture for computational systems that aim to perform Natural Language Processing (3). It provides a common representation for a document and its meta-data, which can be shared across components. It is the foundation of several repositories of tools supporting biomedical text mining, such as `bionlp.org` and `u-compare.org`.

Cross-references

Applied Text Mining

References

1. K.B. Cohen, Karin Verspoor, Helen Johnson, Christophe Roeder, Philip Ogren, William Baumgartner Jr., Elizabeth White, Hannah Tipney, and Lawrence Hunter. High-precision biological event extraction: Effects of system and data. *Computational Intelligence*, to appear.
2. Hong-Jie Dai, Po-Ting Lai, and R.T.-H. Tsai. Multistage gene normalization and svm-based ranking for protein interactor extraction in full-text articles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):412 –420, Jul 2010.
3. David Ferrucci, Adam Lally, and Karin Verspoor, editors. *Unstructured Information Management Architecture (UIMA) Version 1.0.* OASIS Standard, March 2, 2009.
4. J. Hakenberg, R. Leaman, Nguyen Ha Vo, S. Jonnalagadda, R. Sullivan, C. Miller, L. Tari, C. Baral, and G. Gonzalez. Efficient extraction of protein-protein interactions from full-text articles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):481 –494, Jul 2010.
5. Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6 Suppl 1, 2005.
6. Lawrence Hunter and K. Bretonnel Cohen. Biomedical language processing: what's beyond PubMed? *Molecular Cell*, 21:589–594, 2006.
7. Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of BioNLP'09 shared task on event extraction. In *BioNLP 2009 Companion Volume: Shared Task on Entity Extraction*, pages 1–9, 2009.
8. Martin Krallinger, Alex Morgan, Larry Smith, Florian Leitner, Lorraine Tanabe, John Wilbur, Lynette Hirschman, and Alfonso Valencia. Evaluation of text-mining systems for biology: overview of the second BioCreative community challenge. *Genome Biology*, 9 (Suppl 2):S1, 2008.
9. Florian Leitner, Andrew Chatr-aryamontri, Scott A Mardis, Arnaud Ceol, Martin Krallinger, Luana Licata, Lynette Hirschman, Gianni Cesareni, and Alfonso Valencia. The FEBS Let-

ters/BioCreative II.5 experiment: making biological information accessible. *Nature Biotechnology*, 28:897–899, 2010.

10. Preslav Nakov, Ariel Schwartz, Brian Wolf, and Marti Hearst. Supporting annotation layers for natural language processing. In *ACL 2005 Poster/Demo Track*, 2005.
11. Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

View publication stats