ayfie

BOOSTING RELEVANCE WITH LANGUAGE TECHNOLOGY

Search without linguistics – search without sense

Search technology has come a long way since the early days of web search. Open source implementations of indexing algorithms and platform distribution are getting better and better and have already reached a level of excellence that makes them suitable for almost any retrieval application – from enterprise document search to web research engines. Consequently, more and more businesses turn to consultancy companies for custom-made search applications based on their specific content and goals.

Thus, all companies use an adequate search experience that fits both the data and the users perfectly. Or do they? The actual user experience with these search engines is hardly better than the standard from the 1990s and early 2000s. How is that possible?

The answer lies in a misunderstanding about what a search engine as such is capable of and supposed to be doing. This document explains some of the core technologies that can help build a better search experience.

Johannes Stiehler

Chief Technology Officer, ayfie Group AS

Version 3

Dealing with variants

At the core of every retrieval task lies the attempt to match the user's information need to the data available. The prevalent search implementations mostly reduce this task to matching tokens entered by the user to tokens in the textual content. This is based on two assumptions: One, that a user is willing to explicitly state his informational need in a few words (or always capable to do so). Two, that matching those words with words in the content will in fact retrieve the best answers.

Essentially, both these assumptions are wrong:

- The informational need will never be fully specified because there are many implicit contexts that are obvious to the user. They are in fact so obvious to him that he does not even think of mentioning them, such as his location, prior searches etc. Also, the user typically does not even know what content to expect as a result. Often, he will be intentionally vague in order to narrow down the results up front.

- Matching these meager search terms to the exact same terms in the documents rarely produces only the relevant and all the relevant results. This is because it does not consider the many different forms the same concept can take in human language.

Consider the following query:

```
<caucus result>
```

Obviously, the user is interested in the outcome of this specific vote in the US presidential primary. However, we do not know which caucus he is referring to. More likely than not it will be the caucus closest to the asker in either time or location, i.e. the most recent one or the one in his own state.

Now consider a document containing the sentence:

> *"Donald Trump coasted to victory in the Nevada caucuses last week."*

In a stock search engine, this document will never match the query given, firstly because it does not even contain the query term `<result>`, secondly, because "caucuses" is an inflectional form different from the query term `<caucus>`.

It is obvious that to retrieve relevant results, a search engine must deal with these variants effectively. The only precise solution for these types of problems is codified knowledge about human language in the form of electronic dictionaries. This knowledge can then enrich the original text with all variants relevant for retrieval, such as:

- inflectional forms / base forms
- synonyms
- acronym expansions
- superordinate concepts

Thus, the original sentence in the example above would become something like this:

This allows a far broader matching of the user's search terms against the document content.

1.1 Morphology

ayfie builds on more than 30 years of experience in compiling large scale electronic dictionaries and other linguistic resources. Inflectional forms, synonyms and other phenomena (such as decomposition in Norwegian and German) are handled for all major European languages.

These dictionaries are compiled into space-efficient finite state automata that can scan through large amounts of text, annotating all words with their base forms and word segments in the case of compositional forms.

In a search setting, this technology is usually applied both on the document and the query side.

The morphology plugin is available for Solr and Elasticsearch.

1.2 Number normalization

A special case of variant, which is especially important in the case of book search, stems from the fact that all numbers can be represented in several different forms. For instance, a number can occur as a sequence of digits ("100"), several number words or combinations thereof

("hundred", "a hundred", "one hundred") as ordinal ("hundredth") or in some languages even in combination with other words ("hundertjährig", "100jährig" => "hundred-year-old").

ayfie's NumberFilter plugin is able to recognize these different variations and reduce them the digit representation. Thus, no matter how the user enters a number or how it is represented in the text, a match can always be made.

The number normalization plugin is available for Solr and Elasticsearch.

1.3 Semi-automated synonyms

Contrary to common intuition, there are very few "strict synonyms" in human language – i.e. sets of word that can be used interchangeably in any context while retaining the same meaning.

Most synonyms are domain-dependent and context-specific and therefore need to be compiled for each use case.

ayfie supports a semi-automated process for deriving synonyms (and other related terms) by analyzing the content and the query logs. The results of this process can be reviewed by human editors and can then be used directly in Solr and Elasticsearch using the usual document-side synonym facilities.

1.4 Acronym expansion

ayfie comes with dictionaries of acronyms and their expansion forms that can be used to increase recall in a search setting. Acronyms usually either have to be filtered for a specific use-case or applied in a context-sensitive manner.

ayfie's acronym resources come in the form of configuration files for Solr's and ElasticSearch's synonym analyzer.

1.5 Entity Extraction

ayfie supports entity extraction out of the box for person names, locations, organizations, job descriptors and many others. In addition to annotating or extracting these entities, entities can be enriched with semantic markers, for instance to build hierarchical faceting. Hence, ayfie cannot only detect a company name such as "Microsoft", but also knows its legal form and stock trading symbol and can map all those variants to the same entity.

1.6 Terminology extraction

Every domain has its own vocabulary of specific terms that are used to designate important concepts. These terms are usually not single words but rather sequences of nouns, e.g. "breast cancer", "suffix tree" or "revenue sharing agreement".

Identifying those terms and normalizing them enables cross-document summarization as well as much higher precision in many common information retrieval tasks such as clustering and classification.

ayfie's terminology extraction ensures that only truly salient concepts are retained and that most variations in expressing those concepts are properly dealt with.

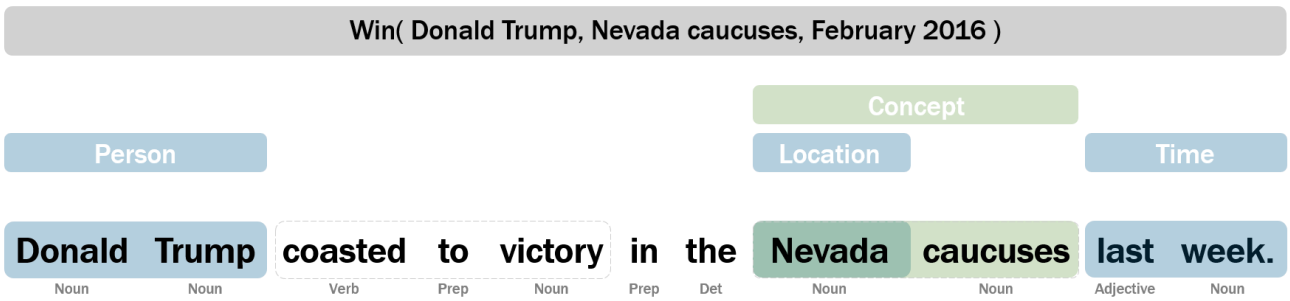
Entity and terminology extraction are based on a local grammar services that runs as a standalone application. Off the shelf integrations exist for both Solr and ElasticSearch.

1.7 Semantic search

Based on the aforementioned annotations, ayfie also enables the detection, extraction and normalization of arbitrary predicate-argument structures through the use of local grammars.

Thus, it is possible to map all the different ways human language can express a logical proposition into a machine-understandable formalized pattern.

This can then be used to create tabular structures, aggregations, time series or enable truly semantic search when done on both the document and the query side.



Contact

For more information please do not hesitate to contact us at the addresses given below:

Rob Wescott · rob.wescott@ayfie.com
Chief Strategy Officer, ayfie Inc.
www.ayfie.com