Natural-Language Text Understanding: The Critical Enabler for Analytics

TeraCrunch White Paper

November 2015

Author: Naveen Ashish, PhD
Co-Author: Kevin Payne, PhD

Introduction

This paper discusses techniques, tools and software frameworks for information extraction and synthesis from unstructured, natural language text. We see a significant component of potentially useful data that is unstructured, semi-structured, or free (natural-language) text in multiple TeraCrunch data analytic applications ranging from predictive analytics forecasting to customer care and response and in verticals ranging from finance to health care to retail.

A particular focus is on information extraction in support of applications such as information retrieval and search. There is a resurgent interest in the fields of text *mining* or *text analytics*, which has been driven by applications such as information retrieval and search, predictive analytics and data mining over natural language text data in a variety of domains and media. There is a virtual explosion of natural language text data in modalities such as social-media and networking forums, internet discussions boards, the Internet overall, and also text content generated from applications in mobile devices. There is interest in a wide variety of domains ranging from consumer health advertising and e-commerce, to intelligence, education and other areas in being able to tap into the information and insights in the unstructured data in these domains.

To be able to glean insights over unstructured data or employ it for analytics, the data must be first converted to a structured representation. This structured data can comprises of the following kinds of elements:

- **Explicitly** mentioned elements in the text, also called named-entities such as person or organization names, domain specific entities such as medication names, specific emotions in the text, etc.
- **(More) Abstract** messaging in the text, such as sentiment, expressions or intent which are not expressed using an explicit word or phrase but are rather implied by the text overall.

Information extraction is the process of synthesizing such elements from natural-language text thus constructing a structured representation of the text. We are interested in determining such elements from individual text units - for instance an individual Facebook post, or a single Tweet and we are also interested in mining information at the corpus level i.e., over a large collection of individual text units. An instance of the latter would determining topics and trends in a large collection of news stories.

In this report we provide an overview of key information extraction methodologies and the application to information retrieval and search.

Information Extraction and Synthesis

Before any actual information extraction from or classification of the text, one can conduct a variety of text pre-processing and analysis operations. This results in representations and information in the text that then then be used for extraction or more abstract classification. This section discusses the text pre-preparation and actual information extraction and synthesis phases separately.

Preparing Text for Information Extraction

We discuss some key preparatory analyses on the text starting from the data in its raw form.

Raw Text "Normalization"

This initial step is important in terms of "cleansing" any data before analysis. Here we typically remove unwanted characters or symbols, we could differentiate between actual words vs punctuations or other emoticons (common in social-media text). We could further normalize the text in terms of case. Finally, in the case of Web data we may also employ a simple parsing based on HTML where we segregate a unit of text (such as an Internet discussion thread or post) into its constituents such as the (post) title, the actual content, author name, date information etc.

NGram Analysis

An ngram is nothing but a sequence of tokens in the text. It is useful to identify ngrams such as unigrams, bigrams or trigrams that are frequently used in a collection of text. The frequent ngrams help us identify commonly used words and phrases in a particular domain and application, and often can be the basis for designing lexicons or ontologies to help with information extraction. The gram *model* is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (n − 1)–order Markov model. ngram models are now widely used in statistical language analysis as well as in information extraction.

Ngram analysis can be conducted on text with words and terms in their raw form or also where the words and tokens have been stemmed to their root words. In the case we are bucketing the text into categories or *aspects*, one can do ngram anaylsis combined with analysis such as TFIDF. This helps us in identifying discriminative ngrams i.e., ngrams that can help classify text to a particular category or aspect.

Tools

Some useful ngram mining and analysis tools include:

- http://homepages.inf.ed.ac.uk/lzhang10/ngram.html
- https://pythonhosted.org/ngram/
- http://code.google.com/p/word2vec/

Part-of-speech Tagging

Part-of-speech (POS) tagging provides the POS tags for all tokens for all sentences in a given segment of text. The Penn Treebank[1] defines 36 distinct POS tags. In corpus linguistics, POS tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context—i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph.

POS tag information can be employed for enabling a variety of information extraction capabilities. For instance for sentiment or emotion classification it is typically the adjectives in text that are most useful. On the other hand nouns or noun phrases would be more useful in classifying the text to a category or the aspect of what it is about (for instance if a comment is about 'food' vs 'service').

Once performed by hand, POS tagging is now done in the context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags. POS-tagging algorithms fall into two distinctive groups: rule-based and stochastic. There is also work on employing machine-learning classification approaches for POS tagging.

[1] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Tools

- TeraCrunch Socratez Text Analytics Engine is built on Stanford CoreNLP library[2] that includes a PosTagger module that can identify POS tags with high accuracy. Several other tools are available with similar functionality, including toolkits in Python.

Natural-Language Parsing

The information retrieval requirements determine the depth to which we need to analyze the text and sentences therein. In some cases we may indeed want to actually parse (at least some of) the sentences in the text. A sentence parse results in what is a called a parse tree which is a hierarchical semantic representation of a sentence.

The information in a sentence parse be use for multiple capabilities. It breaks sentences into clauses and phrases and often we are interested in only particular kinds of clauses or phrases in a sentence. Further, a sentence parse is very useful in establishing relationships between concepts. In fact parsers also provide a relational representation of a sentence parse i.e., it provides the key concepts resulting from the sentence parse as well as explicit relationships between pairs of concepts in the sentence.

Tools

- TeraCrunch Socratez Text Analytics Engine is built on Stanford CoreNLP library mentioned earlier that includes comprehensive sentence parsing capabilities.

Ontology Based Concept Identification

In many applications, particularly in more specialized domains, the text contains or refers to specific terminology for that domain. For instance text in a medical or health domain will contain references to concepts such as drugs or medications, procedures, particular health organizations, symptoms and side effects etc. In such domains it is useful to also conduct an ontology driven analysis or overlay over the text. A simple but useful analysis is to identify terms from one or more specified ontologies in the text.

Such an ontology driven semantic overlay essentially anchors terms to ontology concepts. The benefits are multiple. First, we are able to anchor terms and tokens to actual concepts that are defined, understood and shared by others. Next we are able to normalize multiple representations of the same term to a unified concept - for instance we would be able to

[2] http://nlp.stanford.edu/software/corenlp.shtml

anchor all references such as “L.A.”, “Los Angeles”, or “City of Angels” to a unified concept for ‘Los Angeles’ in a (locations) ontology.

Ontology overlay can be done “top-down” i.e., we take predefined concepts and the ways they can be represented from the ontology and search for the representations in the text. We can also do this in a “look-up” fashion where we consider particular words and phrases from the text and attempt to match them to representations of concepts in the ontology.

Tools

In many applications the ontology overlay capabilities are developed ground up. However there are a few tools available for this task, namely:

- The Apache UIMA Ontology/Dictionary Mapper [3] which requires an ontology provided in a specific XML format and when can then provide a semantic overlay given text input.
- MetaMap [4] from the National Library of Medicine which is also a semantic overlay tool but specific to the medical informatics domain and to the “UMLS” medical ontology.
- TeraCrunch Socratez Text Analytics Engine includes several generic ontologies built in to quickly add additional domain specific ontologies.

Extraction and Synthesis

The pre-preparation analysis prepares the text for the extraction and synthesis of elements that we actually want to distill from the text. We now discuss the key elements that form the structured representation.

Named Entity Extraction

Named entities are of fundamental importance in most domains and applications. Named entities are typically restricted to those entities for which one or many rigid designators, stands for the referent. Peron, organization, location and other names are important kinds of named entities.

Full named-entity recognition is often broken down, conceptually and possibly also in implementations, as two distinct problems: detection of names, and classification of the names by the type of entity they refer to (e.g. person, organization, location and other). Temporal expressions and some numerical expressions (i.e., money, percentages, etc.) may also be

[3] https://uima.apache.org/
[4] http://metamap.nlm.nih.gov/

considered as named entities. With such a reference such as '2015' may be unambiguous in that it refers to the 2015 calendar year, whereas a reference such as 'this Friday' is less distinct. Named-entity recognition algorithms employ linguistic grammar-based techniques as well as statistical models, i.e. machine learning. Hand-crafted grammar-based systems typically obtain better precision, but at the cost of lower recall and months of work by experienced computational linguists. Statistical NER systems typically require a large amount of manually annotated training data. Semi-supervised approaches have been suggested to avoid part of the annotation effort. Many different classifier types have been used to perform machine-learned NER, with conditional random fields being a typical choice.

Tools

- TeraCrunch Socratez Text Analytics Engine is built on Stanford CoreNLP library also includes a named-entity tagger that can identify eight kinds of common entities including person names, organizations, locations and others. Tools with similar functionality are also available in Python.

Location

Identifying location references in text is required in many applications, and in particular mobile applications which typically have a geospatial and location oriented context. Named-entity recognition is one technique that can be employed for location extraction. However one can also employ an ontology driven approach where location identification is done by spotting location references using a pre-defined ontology of locations of interest. The ontology driven approach is useful if we are interested in not just identifying location references but also anchoring them semantically. For instance, using the ontology approach we could not only identify 'Los Angeles' as a location but anchoring it to an ontology concept gives us the additional knowledge that 'Los Angeles' is a city which is part of 'California' (a state) which in turn is part of the 'United States' (a country).

Tools

- TeraCrunch Socratez Text Analytics Engine is built on Stanford CoreNLP library includes a location tagger as part of the named-entity recognizer.
- For the ontology based approach, location ontologies such as from Wikipedia or the 'TGN Locations Ontology' prove useful.

Sentiment

The synthesis of sentiment from text is a ubiquitously desired feature across multiple text search applications and domains. Sentiment extraction and synthesis can be done in multiple ways. The knowledge based approach is a common approach which is applicable across domains. The sentiment extraction is powered by lexicons of what are called polar words. Extensive lexicons of positive and negative polar words are available in English and other languages as well. The sentiment extraction function itself considers the presence and frequency of these polar words (positive and negative) in a segment of text and computes a polarity score

In specialized domains such as health or finance we may need context sensitive polarity lexicons. Contextual sentiment is also very important. For instance consider a statement such as "the wait time is huge". In themselves neither "time" or "huge" have any particular polarity. However it is the combination that results in a negative sentiment. The Deep Learner from Stanford handles this. Most sentiment prediction systems work just by looking at words in isolation and "averaging" over positive and negative words. That way, the order of words is ignored and important information is lost. The deep learning model actually builds up a representation of whole sentences based on the sentence structure. It computes the sentiment based on how words compose the meaning of longer phrases. This results in a more sophisticated sentiment classifier which can learn that 'time' and 'huge' are individually neutral but the combination is typically negative. The underlying technology is based on Recursive Neural Networks (RNN) that builds on top of grammatical structures.

The machine-learning classification approach overall is applicable to sentiment classification. Naïve-Bayes classifiers have been demonstrated to successfully classify sentiment in many domains. One may also employ more sophisticated classifiers with richer feature sets – the feature can include polar word presence or frequency, and features based on the presence or frequency of exclamation marks, punctuations, emoticons, associated ratings etc.,

Conditional-Random-Fields (CRFs) also lend themselves well for sentiment classification, particularly for contextual sentiment classification. Contextual sentiment classification is possible if a classifiers can combine multiple elements in the text towards a sentiment. CRFs are appropriate in that individual elements are captured by labels, whereas the combination is captured by the label sequence model of CRFs. Recent work is also towards employing Neural

Networks for sentiment classification, particularly hierarchical neural networks as Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN).

Tools

- A common approach to sentiment determination is to employ polar word lexicons i.e., dictionaries of positive and negative polar words (for a language). Sentiment is computed based on the presence and also frequency of such polar words within the text. For more advanced sentiment analyzers, word intensifiers (such as very, extremely, hardly etc.) may also be considered.
- Stanford Sentiment Deep Learner[5] is a Java based sentiment classification tool based on Deep Learning.

Emotion

Closely related to sentiment is emotion. The knowledge of emotion from text may be critical, in addition to knowing the sentiment. For instance it may not just suffice to know that an expression has negative sentiment, rather it would actually help to know if the negative sentiment is out of anger, grief, frustration etc.,

The following table, based on a wide review of current theories, identifies and contrasts the fundamental emotions according to a set of definite criteria. The three key criteria used include: 1) mental experiences that have a strongly motivating subjective quality like pleasure or pain; 2) mental experiences that are in response to some event or object that is either real or imagined; 3) mental experiences that motivate particular kinds of behavior. The combination of these attributes distinguish the emotions from sensations, feelings and moods. Robert Plutchik created a *wheel of emotions* in 1980 which consisted of eight basic emotions and eight advanced emotions each composed of two basic ones.

Emotion extraction is typically done using ontologies. We have an ontology or taxonomy for emotions and their variants based on intensity. We further have lexicons of adjectives associated with each emotion.

There are ontologies for emotion that have been constructed.

Tools

Emotion can be determined use ontologies of emotions along with instances of words and phrases attached to specific emotions. Some useful emotion ontologies include:

- http://code.google.com/p/emotion-ontology/
- http://www.obofoundry.org/cgi-bin/detail.cgi?id=MFOEM

[5] http://nlp.stanford.edu/sentiment/

Expressions

The classification of expressions in text was introduced in the Smart Health Informatics Platform (SHIP). The work initiated in the context of health for expressions such as Personal Experience, Advice, Information, Question in health related conversations. This was then broadened to enterprise social-media particularly for enterprise retail with interest in expressions such as complaints, suggestions, questions, acknowledgments, announcements etc.,

This is a feature driven classification problem based on text features such as polar words presence (or frequency), particular POS tags (presence/absence), and words and patterns from lexicons. This also lends itself well to Conditional Random Field (CRF) classifiers with labels for different elements that together define an expression. Expressions are typically domain and also application specific. For instance expressions of complaints for a grocery or restaurant customer comments forum would be different from expressions of say suggestions or complaints for a consumer bank. Thus it is also useful to define features based on domain and application specific lexicons of commonly used words and phrases (for the domain and application).

Corpus Analysis: Topic modeling

The above elements typically apply to individual units of text, for instance they are applicable to a single social-media post or comment, a single news story (document) or a tweet. We are also interested in synthesizing information at the aggregate i.e., over an entire collection of multiple such units of text. A powerful, unsupervised, analysis is that of topic modeling i.e., uncovering topics in a collection of text. In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently: "republican" and "democrats" will appear more often in documents about politics, whereas "score" and "match" will appear in documents about sport. A document typically concerns multiple topics in different proportions. A topic model captures this intuition in a mathematical framework, which allows examining a set of documents and discovering, based

on the statistics of the words in each, what the topics might be and what each document's balance of topics is.

Latent Dirichlet Allocation (LDA) The LDA model considers each document as a mixture of various topics. For example, an LDA model might have topics that can be classified as POLITICS_related and HEALTH_related. A topic has probabilities of generating various words, such as president, capitol, senate, which can be classified and interpreted by the viewer as "POLITICS_related". The HEALTH related topic likewise has probabilities of generating each word: medical, insurance, and hospitals might have high probability. A topic is not strongly defined, neither semantically nor epistemologically. It is identified on the basis of supervised labeling and (manual) pruning on the basis of their likelihood of co-occurrence. A lexical word may occur in several topics with a different probability, however, with a different typical set of neighboring words in each topic. Each document is assumed to be characterized by a particular set of topics. This is akin to the standard bag of words model assumption, and makes the individual words exchangeable.

Tools

- The Mallet Toolkit[6] provides implementations of several topic modeling algorithms including LDA, Probabilistic Latent Semantic Indexing (PLSA) and Pachinko Allocation Model (PAM).

Extraction Elements Summary

Table 1 summaries the information extraction elements discussed above along with the utility or applicability of individual elements as well as the approach and tools for extraction.

Table 1. Information Extraction Elements

Element	Applicability	Techniques and Tools
Named entity	• Broadly applicable in most domains and applications	• Available named-entity recognizers that can be employed "as-is" for common name entities • Ontology driven approach in case the entities are domain specific and have to be anchored and/or normalized
Location	• Broadly applicable and particular in mobile applications where location	• Location references can be determined using named entity recognizers "as is" • Ontology driven approach helps anchor

[6] http://mallet.cs.umass.edu

	and/or navigation is key	the location in a taxonomy of locations
Sentiment	• Broadly applicable in almost any application and domain	• Driven by polarity lexicons • For more advanced sentiment understanding, such as contextual sentiment employ classification approaches such as based on CRFs or Deep Learning
Emotion	• Required (only) in more specialized applications such as customer support, where the sentiment must be further qualified by specific emption	• Ontology driven approach
Expression	• Only required if a deeper understanding of the text is necessary. Relevant for applications such as health informatics, customer support and others	• An abstract classification problem that requires feature driven classifiers • Also certain features recognized by domain specific lexicons
Category/Aspect	• Common feature in many applications, in determining what the "text is about	• Classification approach • Also powered by dictionaries or ontologies
Topic	• Required for aggregate i.e., corpus level analysis of text	• Unsupervised algorithms such as LDA, in toolkits such as Mallet and others

Search

As mentioned above, a key application for unstructured data is that of search. When unstructured data has been comprehensively structured by information extraction, it enables significantly more powerful search and retrieval as opposed to keyword or link-driven search. We discuss semantic search, and faceted search capabilities in particular.

Semantic Search

Semantic search *"seeks to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the searchable dataspace, whether on the Web or within a closed system, to generate more relevant results".* Rather than using ranking algorithms such as Google's PageRank to predict relevancy, semantic search uses semantics, or the science of meaning in language, to produce highly relevant search results. In most cases, the goal is to deliver the information queried by a user rather than have a user sort through a list of loosely related keyword results.

Semantic search systems consider various points including context of search, location, intent, variation of words, synonyms, generalized and specialized queries, concept matching and

natural language queries to provide relevant search results. Major web search engines like Google and Bing incorporate some elements of semantic search. Information extraction provides the capabilities for semantically indexing the text. As opposed to just indexing keywords, a semantic search system can index concepts. Then, retrieval of information can be facilitated over concepts thus enabling better semantic reasoning, synonym resolution, category matching and overall a better match to the "intent" of the search user.

Faceted Search

Faceted search, also called faceted navigation or faceted browsing, is a technique for accessing information organized according to a faceted classification system, allowing users to explore a collection of information by applying multiple filters. A faceted classification system classifies each information element along multiple explicit dimensions, called facets, enabling the classifications to be accessed and ordered in multiple ways rather than in a single, pre-determined, taxonomic order.

Facets correspond to properties of the information elements. They are often derived by analysis of the text of an item using entity extraction techniques or from pre-existing fields in a database such as author, descriptor, language, and format. Thus, existing web-pages, product descriptions or online collections of articles can be augmented with navigational facets.

Faceted search is a natural mechanism for browsing and querying unstructured data that has been structured using information extraction. In particular, if the entities have been extracted as concepts and have been anchored to one or more ontologies. The ontologies themselves can then form the basis for taxonomic filtering, much like as seen in e-commerce sites where one can narrow a (product) search by age group, gender, price range, product category and sub category.

Socratez™: Next Generation Text Understanding

TeraCrunch has developed a next-generation text understanding engine for addressing the key required text analytics capabilities above. This engine, called Socratez™, is a product of leveraging multiple many years of R&D in the areas of semantics and ontologies, natural language processing and artificial intelligence. The engine has a solid core of text analytics

capabilities as a generic framework and is also quickly customizable and optimized as it is applied across multiple kinds of applications and multiple different domains[7].

Conclusion

Information extraction provides the mechanism of converting unstructured natural-language text to a structured and more semantic representation. The structured representation is the basis for significantly more comprehensive data analytics and insights in multiple TeraCrunch applications.

[7] The Socratez™ text understanding engine is described in more detail in a separate TeraCrunch white paper that is also available.