MonkeyLearn Blog | Blog

Create new value from your data

Subscribe

The Definitive Guide to Natural Language Processing

A computer would deserve to be called intelligent if it could deceive a human into believing that it was human. Alan Turing.

When I was a child, I read the book The analytical Engine – Computers Past, Present and Future, by Jeremy Bernstein. Among fascinating concepts such as Turing Machines or the Pascaline, Bernstein mentions machine translation as one of the tasks that a computer could perform. Just imagine that for a second: a machine capable of *understanding* the language of a text and having the required *knowledge* to translate it into another language.

A machine capable of understanding a text like we humans do? I know, it seems hard to believe, and for years it seemed like magic to me, until I discovered Natural Language Processing (NLP), a field that deals with this kind of problem.

Yes, by combining the power of artificial intelligence, computational linguistics, and computer science, NLP allows a machine to understand natural language. A task that, up until now, only results at the time were not especially good and for decades the following example was used to illustrate the limits of the field:

The Biblical saying "The spirit is willing but the flesh is weak" was translated into Russian, and then translated back in English, resulting in "The vodka is good but the meat is rotten". Well, this is just a story, mainly inspired by the flaws of literal translation, but it was so often considered as an actual example that it discredited the NLP field for years.

Since the 1950s a lot of progress has been made and NLP became a major force, in both economic and social development, when the Internet went from being a small neighborhood of friendly computers to a huge repository of exploitable data.

In this guide I'll briefly introduce you to the field, trying to explain some everyday applications of NLP. It is not my purpose to provide an in-depth explanation, nor introduce you to algorithms and techniques. This is just an overview of a field that has fascinated me for so many years.

Use Cases of Natural Language Processing

NLP is everywhere, even if we don't know it. Although the term is not as popular as Big Data or Machine Learning, we use NLP every day. Here are some examples of how NLP is widely used:

Machine translation

Maybe you have already used machine translation and it seems a natural feature to you by now. The *globe* icon on Twitter or the *translate* links in Facebook posts, in Google and Bing search results, in some forums or user review systems.

Machine translation works very well in restricted domains, that is when the vocabulary and the idiomatic constructions are mainly known. It can, for example, significantly cut the costs when it comes to translating technical manuals, support content or specific catalogs.

With advances in NLP over recent years, machine translation is becoming more accurate. Now, machines can even detect text in images and deliver translations.

Automatic summarization

Together with machine translation, automatic summarization was addressed in the 1950s. Given a text, the goal is to get a reduced version of it that retains the most important information. A summary can be created by *extraction* or *abstraction*.

sentence in the summary can refer to elements of a sentence that is not in the summary, a phenomenon known as dangling anaphora:

> *'Volkswagen's new CEO Matthias Mueller has his work cut out for him. Mueller has spent most of his career at the Volkswagen group so he knows the inner workings of the company.* ***Now experts say he'll have to make some big, bold changes to get the largest automaker in the world back on track****.'*

In the example above, if only the third sentence is retained by the system, the reader will certainly ask 'who is the *he?'*.

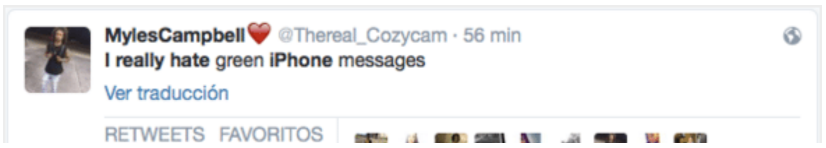
On the other hand, an abstraction-based approach implies text generation: the summarizer does not copy text from the input but writes in its own words what it understands from the text. This is extremely complex and currently most of the systems available uses an extraction-based approach.

Automatic summarization is of particular interest when it comes to delivering an informative overview of a set of journals. Some approaches, called multi-document summarization, can condense multiple documents into one summary. Here it is important to avoid redundancy coming from different documents and maximize diversity. Some other approaches model a summary as a simple list of keywords, so they work more like a keyphrase or keyword extractor. You may have seen systems like blog platforms or scientific literature reference managers suggesting you the most important concepts of your article or paper.

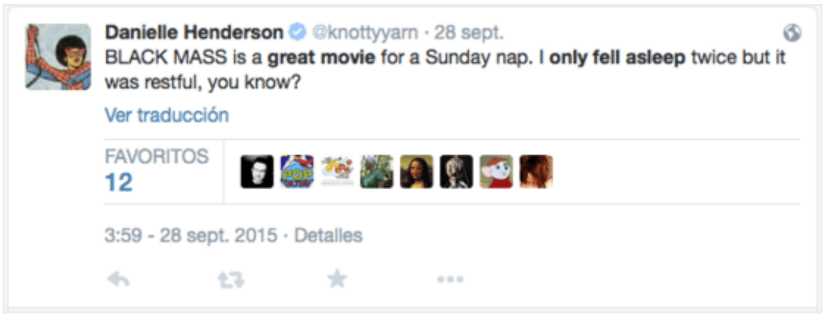
Sentiment analysis

The goal of sentiment analysis is to identify subjective information in texts. It can be a judgment, an opinion or an emotional state, and it is a major issue lately for companies and famous people that want to be aware of their reputation on the Internet. What do users say about our products? What do they think about my restaurant or my hotel? Are they happy with our customer support service? What do they think about the competition?

The most frequent kind of sentiment analysis performed is called polarity detection, that is, understanding if a text about a given subject is positive, neutral, or negative. This might seem simple, and in some cases it is:


MylesCampbell❤️ @Thereal_Cozycam · 56 min
I really hate green iPhone messages
Ver traducción

Clear enough, don't you think? Twitter is an incredibly rich source of information for sentiment analysis. Real time, mainly subjective, easily shareable information. However, sentiment analysis has to deal with some very difficult problems. If you are building a system for a tech company, a user saying that his new cellphone is *light weight* is with no doubt a good thing. But what if what is light weighted is a political debate? Even with words that are likely to be unambiguous, we could face some serious problems:

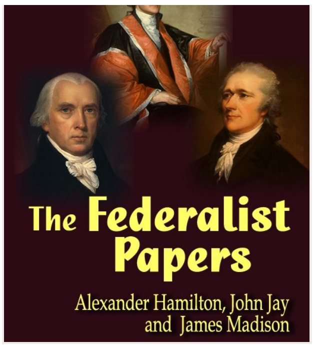

Danielle Henderson @knottyyarn · 28 sept.
BLACK MASS is a **great movie** for a Sunday nap. I **only fell asleep** twice but it was restful, you know?
Ver traducción
FAVORITOS
12
3:59 - 28 sept. 2015 · Detalles

Text classification

The goal of text classification is to assign predefined tags to a text. In some applications, such as automatic spam detection there are only two tags: *spam* and *ham*. We call such an application a *binary classifier*. In other cases, a classifier can for instance organize news stories by *topics* or academic papers by *domains*. Since we have more than two tags , we call it a *multi-class classifier*. And what happens if a blog post talks about *sports* and *entertainment*? How does a classifier choose the correct tag between multiple choices? Well, it depends on the application: it can simply take the most likely candidate, but sometimes it makes sense to assign several tags to a text. That is called a *multi-label* classification problem.

Not all the applications in text classification are semantic oriented. For example, we can use a classifier to perform automatic language detection, that is, to detect in which language a text is written.

Some applications are particularly sophisticated. One that I love is authorship attribution: based on the *style* of a text, a system must decide, with a reasonable margin of error, its author. Can you imagine that? Such a system must know the set of candidate authors, and for each one of them, it must know a set of text samples. There are a lot of ways to model style, which is in fact a very complex concept. Some typical features used by this systems are word frequencies, vocabulary richness, sentence length and phrase structure. Although this is a current research subject, the first remarkable results were obtained in 1964 by the work of Mosteller and Wallace on the authorship of



The federalist papers.

Conversational agents

Do you remember HAL 9000 from 2001 a Space Odyssey? HAL speaks, thinks, is empathetic, he recognizes faces, express feelings and can even perform lip-reading and lie. A conversational agent (CA) is a system capable of having a coherent conversation with a human. It can present a text-only interface or be a spoken dialog system. Usually the conversation has a precise goal. For example a customer service CA, often called automated online assistant or chatbot, that processes user requests and questions, or a CA that gives traveling advises.

You have probably already interacted with a conversational agent, maybe without knowing at first that it was in fact a machine. I also guess that maybe you have realized during the conversation that something was missing, something human, a certain degree of fluency and unpredictability in the dialogue. The thing is that we are currently far from building such an intelligent entity as HAL. But if we focus on a particular subject or task, where we only deal with some kind of restricted or formulaic language, we can get pretty good results.

In 1964 Joseph Weizenbaum wrote ELIZA, a computer program that simulates a conversation with a Rogerian psychotherapist. Here there is an example of a possible conversation.

> ELIZA: Please tell me what's been bothering you.
>
> YOU: Nothing, I feel really happy today
>
> ELIZA: Tell me more about such feelings.
>
> YOU: Well, I'm just happy
>
> ELIZA: Has your treatment made you happy ?

YOU: I don't want to talk about that

ELIZA: Does that trouble you ?

The conversation can go on like that for hours. ELIZA uses very basic pattern matching techniques, but at the time the impact was enormous. Since then, a lot of improvements have been made. Take a look at what happens when two spoken dialog systems talk to each other.

So many other NLP applications…

There are plenty of other NLP real world applications. Does your email application automatically detect events and propose you to add them to the calendar? Well, that's an NLP task called Information Extraction. Do you use a spell checker that recognizes grammatical errors? Have you already tried a software of handwriting recognition? A question-answering system such as Ask.com or START from the MIT? What about content-based advertising? Apple's Siri, IBM's Watson? Do you know that since 2012 Forbes magazine uses the storytelling system Narrative Science to automatically write online articles? Have you seen lately the huge progress in robotics?

NLP is everywhere even if we don't know it. And although NLP applications can very rarely achieve performances of 100%, they are part of our lives and provide a precious help to all of us.

different aspects of language: phonology, morphology, syntax, semantics and pragmatics. And its worst enemy is called *ambiguity*. We are going to see different levels of analysis (I will set aside phonology since it's less intuitive and needs specific background) and how NLP systems usually address them.

Words in a sentence

The first thing to understand are words, in particular the nature of each word. Is it a noun or an adjective? If it's the inflected form of a verb, what is its infinitive form, and to what tense, person and number correspond the inflected form? This task is called Part-of-Speech tagging (PoS). Let's take a look at the following sentence:

> John bought a book

Well, there is a straightforward approach: we could use a dictionary with the information of all the words, their inflected forms and part of speech, to compute the following output:

> John/ProperNoun
> bought/Verb-past
> a/Det
> book/Noun

Alright, let's set aside the fact that a language is an incredibly rich living entity, thus we can never know all the words. As we can see, even with a simple sentence this approach can't work. The word *bought* can also be an adjective, *book* is a verb and *a* could be the letter, in which case its PoS would be noun. As humans, we can usually resolve this kind of ambiguity. But try with the following sentence:

> Will Will will the will to Will?

Real NLP applications usually perform tasks using two families of approaches: *symbolic* and *statistical*. A symbolic approach consists on a set of rules, often hand-written but sometimes automatically learned, that *model* different language phenomena. A statistical approach typically uses machine learning algorithms to *learn* the language phenomena.

a rule could correct that:

> Adj → Verb if the tag of the previous word is ProperNoun

Statistical approaches see PoS tagging as a sequence labeling problem. The underlying idea is that given a sequence of words with their respective tags, we can decide, for the next word, the most likely PoS. In our example, if we have already seen "John bought a" and we know their PoS tags, we can say for sure that "book" is a noun and not a verb. This makes perfect sense. There are statistical models such as hidden Markov Models (HMM) or Conditional Random Fields (CRF) that can be trained using big corpus of labeled data, that is, texts where each word has the correct PoS tag assigned.

From words to structure

Alright, our NLP application knows how to deal with words. For example, it can find in a text all the instances of the verb *buy*. Next question is: who buys what? We are going to need syntax for that. That's a hard problem, sometimes even for a human. For example, in the following sentence:

> I saw a man on a hill with a telescope

Do I have the telescope or the man has it? Is there a telescope on the hill? Am I on the hill or he is?

As you can see, syntax can be tricky. We have to understand how words group together in units called *chunks*, and how these chunks relate to each other. For that, we use grammars. For example:

> S → NP VP
>
> NP → Det Noun
>
> NP → ProperNoun
>
> VP → Verb NP
>
> ProperNoun → John
>
> Verb → bought

tree is built. We will know then that it's [John] who bought [a book]. Grammars can be manually written or learned from treebanks, that are text corpus annotated with parser trees.

Getting the meaning of words

The humorous adage 'Time flies like an arrow. Fruit flies like a banana.' illustrates perfectly the complexity of semantic ambiguity. There are two main problems:

1. polysemy: words that have several meanings
2. synonymy: different words that have similar meanings

There are other semantic relations such as antonymy, hyponymy or hypernymy, that are important, but polysemy and synonymy are usually the most relevant.

Lexical semantics deals with the meaning of words as units, while compositional semantics studies how words combine to form larger meanings. So there are several approaches to semantics, that remains a major open problem in NLP.

Word sense disambiguation (WSD) tries to identify the sense of a polysemic word in a given sentence. Let's take, for instance, the following sentences:

> The tank is full of soldiers.
> The tank is full of nitrogen.

As you can see, it is really a very hard problem. In both sentences PoS tagging and syntax are the same. How can an NLP application know the kind of tank for each sentence? Well, a deep approach can be used. It requires world knowledge. For example: a tank (container) usually don't hold people. This is of course too costly, so shallow approaches are more frequent: given the word tank in a sentence, what are the nearby words? It makes sense: when we see the word *soldiers* or *nitrogen*, we, humans, can determine the kind of tank. So the co-occurrence of words can be used to remove ambiguity. This knowledge can be learned from a corpus where the meaning of each word is correctly labeled.

Most research in the field of WSD is performed by using WordNet, a very large computational lexicon that groups nouns, verbs, adjectives and adverbs into sets of synonyms called *synsets*. Each synset represents a unique concept. For example, for tank we have five synsets for noun and three for verb. The two synsets related to our example are {tank, army tank, armored combat vehicle, armored combat vehicle} and {tank, storage tank}.

Regarding compositional semantics, things get a little bit more complicated. First, the key idea is the principle of compositionality, that states that the meaning of a whole can be build from the meaning of the parts. In compositional semantics, the parts are usually the constituents of the syntactic parse. So, to understand a sentence, we build a logical representation of it. We could use for that first-order predicate logic, where predicates have specific non-ambiguous meaning. Let's imagine a user that writes the following request:

> a hotel with sea view near San Francisco

To correctly process the request, the system must understand it in a very precise way. So it could build the following logical representation of the query:

> ∃x Hotel(x) Λ View(x, Sea) Λ Near(LocationOf(x), LocationOf(SanFrancisco))

The system could use then a programming language as Prolog to compute the results. This is, of course, very hard and costly in a general case, and global meaning can't always be derived from the meaning of the parts, but in a restricted domain we can get pretty good results. Compositional semantics remains an open problem and it is a current crucial research topic in NLP.

A word about pragmatics

It is not what you say or how you say it, but the context. Pragmatics studies the ways in which context contributes to meaning. Imagine the following dialogue:

> JOHN: You are an idiot.
> PETER: Oh, thanks, that's very kind of you.

Should an NLP application consider that Peter thinks that John's words are kind? And what happens if John was simply joking? Irony and sarcasm are very complex mechanisms. There is research work in NLP that intends to characterize them. For example, we can train a classifier to decide whether a tweet is ironic or not. Features could be things like word frequency (assuming unexpectedness is a signal of irony) or the use of some adjectives (assuming exaggeration is also a signal of irony). But, although there are some interesting results, it is still an open problem and there is much space for improvements.

likely train a Naive Bayes classifier over the words. So basically the application needs to know how to segment text into words, a process call tokenization. On the other hand, complex applications such as question-answering or information extraction systems, would probably need to perform morphological, syntactic and semantic analyses to give decent results.

What Is the best NLP Approach?

Symbolic or statistical? The short answer is neither. There have been a lot of discussions about this, with a non-negligibly number of radical positions. Fred Jelinek, who led the IBM's speech recognition team for about twenty years, reportedly said that:

"Anytime a linguist leaves the group the recognition rate goes up", which is often quoted more directly as: "every time I fire a linguist, the performance of our speech recognition system goes up".

On the other hand, famous linguist Noam Chomsky stated in 1969 that, "it must be recognized that the notion of *probability of a sentence* is an entirely useless one, under any known interpretation of this term".

What can be called the first age of NLP was dominated by symbolic approaches. As from the 2000s, statistical approaches became a reality and over the years even the most critics had to admit that statistical NLP gives very good results and outperforms sometimes the purely symbolic approaches.

The good point about statistical methods is that you can do a lot with a little. So if you want to build a NLP application, you may want to start with this family of methods. I recommend you to read the post about machine learning by Raúl Garreta, that gives some examples of statistical approaches that come usually from the machine learning field.

Statistical approaches have their limitations. When the era of HMM-based PoS taggers started, performances were around 95%. Well, it seems a very good result, an error rate of 5% seems acceptable. Maybe, but if you consider sentences of 20 words on average, 5% means that each sentence will have a word mislabeled. Since then, improvements were made, either by building better training resources or trying other models such as CRF, but there is always an error rate you have to deal with. In some cases, the rate is so slow that the problem can be considered as solved. Automatic spam detection is one of them.

On the other hand, although they are more expensive, symbolic approaches are easier to understand by a human, without the need of a specific background in probability and statistics. Moreover, in some way you can have a better control of the methods: you can delete or change a any case you will have sometimes the impression of dealing with a black box.

So the answer to the question of whether using symbolic or statistical approaches would depend on the applications, the goals, the existing resources, the people's background, the budget, among other variables. You can also try a *hybrid approach*. You can use statistical methods to create automatically the rules (as the Brill's PoS tagger do, for example) and then eventually curate them manually. On the other hand, you can inject manually build knowledge into a statistical method to improve results. For example, you can write a rule to state that a noun group can never end by a determiner. So if your system was trained over a non-curated corpus, this error won't be made.

Final words

Fifteen years ago, it was really hard for an NLP company to tell a prospect that their product doesn't work with absolute precision. Times have changed. With the advent of better search engines, blogs, social networks, e-commerce, the society slowly began to understand that, even perfectible as it is, NLP provides us a *precious* help every day.

NLP is a *young* field, full of promises and with an international community that continues to develop new algorithms, techniques and resources. It moves *fast*. The recent results using Deep Learning are amazingly improving several hard NLP tasks. The combination of advances in artificial intelligence and the emergence of the Internet prompted NLP to a level *unthinkable* just a few years ago.

At MonkeyLearn, we believe that NLP is a *game changing* technology and that it will shape the future of Internet. NLP is a difficult field, yes, and that's exactly what we want to change.

Our goal with MonkeyLearn is to make NLP accessible to everyone and hopefully empower the next generation of intelligent applications. Sign up for free to start your NLP journey.

We will see in the future how far NLP, machine learning and artificial intelligence technologies can go. Sometimes I wonder, will we see the day when machines everywhere would be able to pass the Turing test?

Posts you might like...

Bruno Stecanella • May 10, 2019

Have something to say?

MonkeyLearn Blog | Blog

Subscribe

Create new value from your data

Mechanical Turk 101: How to use MTurk ...

2 years ago • 2 comments

Mechanical Turk 101: How to use MTurk for tagging training data Accurately …

Everything You Should Know about ...

a year ago • 1 comment

Everything You Should Know about Auto-tagging Customer Feedback …

Automatically Tag Conversations in ...

2 years ago • 4 comments

Automatically Tag Conversations in Front using Machine Learning …

Getting sta Natural Lan

2 years ago • 2

Getting starte Language Pro Let's say you

What do you think?

0 Responses

Upvote Funny Love Surprised Angry Sad

0 Comments **MonkeyLearn** **Disqus' Privacy Policy** 1 **Login**

Recommend **Tweet** **Share** **Sort by Best**

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

Subscribe **Add Disqus to your siteAdd DisqusAdd** **Do Not Sell My Data**

Text Analysis with Machine Learning

MonkeyLearn | Blog
Blog

Create new value from your data

Subscribe

Try MonkeyLearn

Clearbit Segment PubNub PROTAGONIST

MonkeyLearn | Blog

- Analyze text at scale with Machine Learning
- Try MonkeyLearn