NATURAL
TECH

SentiLecto:
the model
from inside.

SentiLecto: the model from inside.

Why is it important to know about language and syntax? From a word-based approach that ignores syntax, you only can count word occurrences. It was the first approach in the automatic analysis of text. However, this approach soon displayed its limitations: someone can say very different things using the same words. For example, from this approach, you can't distinguish who surprised everyone in these sentences.

Ex.: *Microsoft compró Oracle y sorprendió a todos.*
(Microsoft buy Oracle and surprise everyone)

Ex.: *Oracle compró Microsoft y sorprendió a todos.*
(Oracle buy Microsoft and surprise everyone)

Perhaps, seeing these examples, you think that entities order of appearance could be a good hint to solve this kind of problems. Maybe this is true for the English language, but in Roman languages (like Spanish, Portuguese and many others) constituency order is much freer.

Instead, a syntax based approach allows you to access a true representation of the facts expressed in a sentence. It permits you to know who makes what to whom, that is, who makes some action, and who receives its consequences. In addition, a syntax based approach can solve passive voice sentences and subordinated sentences. For example, SentiLecto can tell you who say what and who create what in this example:

Ex.: *Bill Gates dijo que Apple fue creada por azar por Steve Jobs.*
(Bill Gates said that Apple was created by chance by Steve Jobs)

So, full parsing is the best approach in order to perform an accurate text analysis. Following, we will explain how SentiLecto works in order to perform a wide variety of tasks related to text analysis.

SentiLecto is mounted over Freeling, an open source language analysis tool suite. But, before giving an input to Freeling, SentiLecto makes some preprocessing tasks over the entire text input.

PIPELINE:
Preprocessing (paragraph, sentence) -> Freeling -> (clauses) SentiLecto

1) Preprocessing:

In the first place, SentiLecto divides the text input in paragraphs. Why is it important to split the text in paragraphs? There are some linguistic behaviours belonging to the paragraph's scope, essentially referential issues, like anaphora resolution. The paragraph indicates a formal limit to this kind of behaviours with the advantage of being unequivocal, because there is a formal mark that identifies it, the newline “\n”.

Once we divided the text in paragraphs, SentiLecto makes a second preprocessing, it divides each paragraph into sentences. That is important because the Freeling input should be sentences. Instead, we will see later that the SentiLecto input should be clauses.

There are a couple of considerations to do, related to how to encode the sentences formal limits in an algorithm. An approach that only takes into account the period mark (“.”) will result in both false positives and false negatives:

Ex. *El Dr. Román Lipsky recalcó los resultados de sus investigaciones.*
(Dr. Roman Lipsky emphasized the results of their investigations)

Ex. *Compré una nueva heladera, sin embargo, pronto me arrepentí.*
(I bought a new refrigerator, however, soon regretted it)

In the first example we can view how the period mark have other uses as well as identifying sentence formal limits. It is also used in abbreviations, like Dr., Sr., etc.

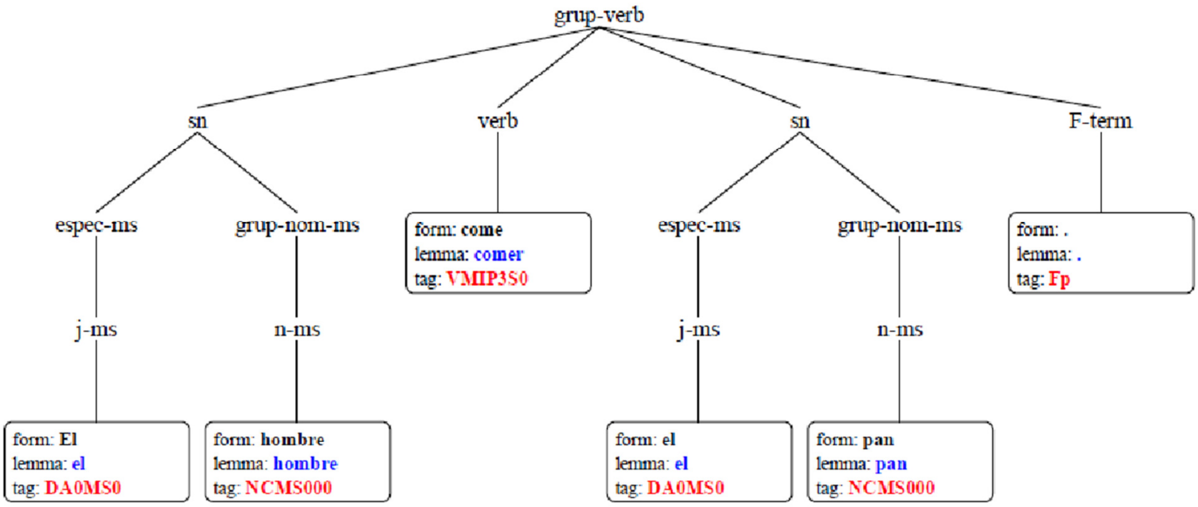
In the other example we showed that, besides the period, there are some words (mostly connectors) that also work as formal limits. So, in the second example there are two sentences: “compré una nueva heladera”, “pronto me arrepentí”, joined by the connector “sin embargo”.

2) Freeling:

Once we have the text divided in a set of sentences, we give each sentence to Freeling as input. Freeling carries out two main tasks: it PoS-tags all the words in the sentence and then, using its context free grammar (CFG grammar), builds a syntactic-tree-shape representation.

The Freeling PoS tagger combines a vast dictionary and a tag guesser, for words that are not in the dictionary, like neologism or certain idiomatic expressions. Although the Freeling syntactic tree is a good first approach in order to build an accurate representation of the meaning, it isn't a real full parsing: its tree doesn't display the existing hierarchy among different structures, which is required, for example, to distinguish between a subject and a direct object.

The following syntactic tree belongs to the Freeling online demo (http://nlp.lsi.upc.edu/freeling). The input was “El hombre come el pan”. In the tree, we can see how the Freeling's output doesn't reflect a hierarchy between the noun phrases (labeled with the tag “sn”). While “el hombre” works as subject, “el pan” works as a direct object and consequently it depends on the verb. However, in the Freeling' output the two noun phrases appear at the same level.



In conclusion, as we have seen, Freeling doesn't perform a real full parsing, but only a shallow parsing, which doesn't allow us to distinguish, for example, between internal and external arguments.

As we said before, English has a much more regular word order than Roman languages. This fact guarantees that the first noun phrase, before the verb, was the subject, and the second one, at verb's right, was the direct object. Roman languages, instead, are free constituent order languages. In consequence, it is very difficult to identify with high accuracy the syntactic functions of each noun phrase that was previously identified by the parser. We refer to this difficult to identify which syntactic function performs each noun phrase as the Structural Subject-Verb-Object (SVO) Problem.

SentiLecto limits the error scope to the clauses, so it has to build this concept, that doesn't appear in Freeling. This analysis unit change, from the sentence to the clause, constitutes a SentiLecto's improvement before Freeling, especially when it comes to perform fact mining. The clause extractor is the resource that performs this task. It identifies which nodes in the syntactic tree (the Freeling output) work as clause separators and it uses them to split a complex sentence into clauses. They are: "grup-verb", "subord-rel", "subord", "subord-ger", "verb-pass".

3) SentiLecto:

SentiLecto solves the Structural SVO Problem, aforementioned above, by building a meaning representation based in slots. These slots are a simplification of the superior syntactic functions. We call them “SVO slots”, because of the main slots: subject, verb and object.

Unlike Freeling, SentiLecto does an entity based parsing, that is, based on entities recognition. It takes the Freeling syntactic tree and analyzes its chunks one by one, in order to fill the SVO slots and identify entities.

Following, we detail each slot:

-Opinionated modal adverb
-Subject
-Verb
-Direct Object
-Indirect Object / Experiencer
-Phrasal Verb Complement
-Equative
-Instrument / Company / Goal / Beneficiary
-Adjuncts (space, etc.)
-Time

But before SentiLecto fills the SVO slots, each verb has to be classified according to its argumental requirements. The reason for this is that different types of verbs require different types of arguments. This information is encoded in a resource called scripts_verbales.csv, which lists all the verbs and details some features for each one. In addition to their argumental requirements, this resource contains information about the animacy requirements for each verb argument and its type of sentiment spilling.

There are six types of verb transitivity: transitive, intransitive, regimen, meteorological, equative, auxiliary verb, and uncommon or rare verbs. The following table details them and includes the identification code used by scripts_verbales.csv.

Types of transitivity		
Code	*Category*	*Description*
`T`	Transitive	This verb type requires a direct object.
`I`	Intransitive	This verb type doesn't select any complement.

R	Regimen	These verbs select a prepositional phrase as their complement.
M	Meteorological	This label is reserved for meteorological verbs
E	Equative	These verbs typically refer to constructions where two entities are equated with each other.
H	*Haber* verb	This label is reserved for the auxiliary verb
X	Uncommon, rare verbs	This label is reserved for uncommon, rare verbs, but in fact it configures them to act like intransitive verbs

For each verb category, there will be a different set of slots available, which we detail following:

- TRANSITIVE: verb, subject, od, oi, instrument/goal, adjuncts.
- INTRANSITIVE: verb, subject, oi, instrument/goal, adjuncts.
- REGIMEN: verb, subject, regimen, instrument/goal, adjuncts.
- EQUATIVE: verb, subject, oi, equative, instrument/goal, adjuncts.
- EXISTENTIAL: verb, od, instrument/goal, adjuncts.
- IMPERSONAL: verb, instrument/goal, adjuncts.

So, in order to build a SVO representation, each entity will be allocated in one of the available boxes. It is important to emphasize that, once the parser assigns an entity to a certain slot, it may not be relocated in another slot. In consequence, there is a priority order in the assignments:

- Subject
- Direct Object
- Indirect Object
- Regimen
- Equative
- Time
- Instruments/Goal
- Opinionated Modal Adverb
- Adjuncts

Following, we are going to explain how SentiLecto proceeds in order to find the entity that fills syntactic function. It combines the information provided by the Freeling syntactic tree with some heuristics of its own.

The two main heuristics are Distance and Levels. Distance is a measure of how far away the algorithm will search for an entity to fill a certain slot. Sometimes, it also specifies and grant a privilege to certain side, mostly the verb's right side, except for the subject extraction. On the other hand, Levels refer to the syntactic hierarchy, and could differentiate, for example between a main clause and a relative clause in a complex sentence.

The usual process to fill a slot is to find candidates, evaluate them and choose the right one. This process is executed by ActiveVoiceSyntacticAnalyzer.py and PasiveVoiceSyntacticAnalyzer.py.

1) Subject extraction: Although Spanish and Portuguese are languages with a free constituent order, there is a canonical form Subject-Verb-Object, which statically is the most common. In consequence, in order to find the subject, SentiLecto will first search for entities on the left side of the verb, and only if it doesn't find it there, it will search on the right side of the verb. Of course, in order to fill the subject slot an entity has to agree in person and number.

When we talked about the types of verb transitivity, we mentioned a resource called “scripts_verbales.csv”, which contains an exhaustive verb list with syntactic and semantic information for each verb. As part of the semantic information that this file contains, it details the animacy requirements for each verb subject. That is because some Spanish and Portuguese verbs necessarily require a certain animacy feature for its subject. This field accepts three options: “anim” (animate) “no” (inanimate) and “bi” (unspecified). For example, all the *saying verbs* (like *decir, gritar, comentar, susurrar, etc.)* necessarily require an animate subject like *vendedor* in the following example.

Ex.: *El vendedor ambulante gritaba a viva voz sus ofertas.*
(The street vendor shouted loudly their offers)

In consequence, when a verb requires a certain animacy feature to fill the subject slot, an entity has to agree in person, number and also in its animacy features. In the next example, we show the use of a verb that requires an inanimate (“no”) subject, “acontecer”:

Ex.: *Un desastre natural aconteció esta mañana: la erupción del volcán Popocatépetl, en México.*
(A natural disaster happened this morning: the eruption of Popocatepetl volcano in Mexico)

The animacy feature is encoded in a set of resources: person.cls, twGent.dat, animal.csv and three list of names: male.txt, female.txt and unisex.csv. When a word is not in any of this resources, it is considered inanimate.

2) Direct Object Extraction: At this point, we have to divide the explanation, because Spanish and Portuguese have different grammatical behaviours.

a) Spanish has an animacy mark in the direct object, consequently its grammatical form will be different depending on the animacy feature of the entity.

Ex.: El gobernador golpeó <u>la mesa.</u>
(The governor hits the table)

Ex.: El gobernador golpeó <u>a su empleado</u>.
(The governor hits his employee)

As we can see in these examples, when the entity in the direct object is animate, the direct object is built with a prepositional phrase headed with the preposition “a”. Instead, when the entity in the direct object is not animate, the direct object is built with a noun phrase, without preposition. This different behaviour depending on the direct object entity animacy is known as the animacy mark in the direct object.

The resource scripts_verbales contains the animacy requirements for each verb object. It accepts the same three animacy options for the subject: “anim” (animate) “no” (inanimate) and “bi” (unspecified).

So, besides agreeing in person and number, the verb and the entity selected will agree in this animacy feature. If the verb requires an animate direct object, SentiLecto will search for a prepositional phrase, instead it will search for a noun phrase. Keeping in mind this, SentiLecto firstly will search the direct object at the verb right side, the most usual place, with a distance minor or equal to three. Otherwise, it will search at left, the least usual place, with a distance minor or equal to two. In last place, it will search for accusative pronouns.

b) Portuguese, unlike Spanish, doesn't have this animacy mark, so the direct object will be always built using a noun phrase. Consequently, SentiLecto firstly will search the direct object on the right side of the verb in a distance minor or equal to three and then it will search on its left side in a distance minor or equal to two. Finally, it will search for accusative pronouns.

3) Indirect Object extraction: The indirect object will be typically made by a prepositional phrase. Consequently, SentiLecto will search for “grup-sp” (the Freeling tag for a prepositional phrase) firstly on the right side of the verb, in a distance minor or equal to three constituents from the verb at the same syntactical level. If it doesn't succeed, it will search on the left side of the verb, but in a distance minor or equal to two constituents. Finally, it searches personal pronouns in dative case.

Note: in Spanish the direct and indirect object can appear duplicated, once as a noun or prepositional phrase, and once as a personal pronoun. SentiLecto can also deal with this particular behaviour. When a direct object appears duplicated it has a special meaning, the spokesperson is trying to emphasize it.

Ex.: Él la quiere mucho a su hija.
(*He loves his daughter very much)*

However, the indirect object could be duplicated without a special emphasis.

Ex.: Le quitó la pelota a Leandro.
(He took the ball from Leandro)

4) Phrasal Verb Complement extraction: We call certain verbs that necessarily require a prepositional phrase "Phrasal verbs complement" or "Regimen verbs".

Ex.: "lidiar <con alguien>", "creer <en alguien>", etc.

So, we search the prepositional phrases ("grup-sp", according to the Freeling syntactic tree label) headed with a set of prepositions ("a", "por", "en", "de", "contra", "con", "sobre") and located at the verb level. Firstly, we search on the right side of the verb, then on its left side, always limiting the search to a specific distance. Once we have found it, we extract the entity inside it to fill the regimen slot.

Ex.: *El director técnico declaró que* ***confía*** *en sus jugadores.*
(The coach said he trusts in his players)

5) Equative extraction: the Equative box is only available for a small set of verbs, called equative verbs. The most important among them is the verb "ser" (*to be).* To extract an entity to fill this box, SentiLecto will search for noun phrases on the right side of the verb.

Ex.: *Cuando se enoja, mi padre* ***parece*** *un ogro.*
(When he gets angry, my father seems an ogre)

6) Opinionated modal adverb: To fill the opinionated modal adverb slot is quite simple. Firstly, SentiLecto looks for adverbs inside the clause under analysis. But to enter inside this slot, an adverb has to meet an additional requirement, it should have a valuated score in sentiwordnet.csv. Sentiwordnet.csv is another SentiLecto's resource. It contains a list of all the words that imply a positive or negative valuation, each one with a score.

Ex.: *Afortunadamente, no se registraron víctimas del accidente vial.*
(Fortunately, no victims of the road accident were recorded)

7) Instrument / Company / Goal / Beneficiary: These syntactic functions are typically built using a prepositional phrase. At this point, it is advisable to remind you that the slots are filled in an specific order, so SentiLecto will search candidates to this slot only after it has found the entities for the DO, the IO and the regimen.
To fill this slot, a prepositional phrase should be headed by a preposition included in a set of allowed prepositions: "sin", "con", "contra", "en favor de", "a favor de", "en defensa de", "en contra de", "en pos de", "mediante", "a través de", "por medio de", "gracias a", "merced a".

Ex.: *El diputado dió una larga entrevista en defensa de las políticas de su partido.*
(The deputy gave a long interview in defense of his party's policies)

8) Time extraction: the time box is reserved to temporal expressions. They could be specific (like *3 de mayo de 1996*) or unspecified (like *dos horas después*).

The TimeDate module combines several heuristics. There is a resource called "Time Words", that lists words related to specific or unspecified temporal expressions, like "septiembre", "invierno", "mediodía", "temporada", "siglo", etc.

In addition, the module includes a list of non temporal prepositions, which can never head a prepositional phrase with a temporal value: "por", "de", "con", "contra", "a", "ante", "bajo", "sin", "según", "tras", "a_causa_de".
Finally, it also works using regular expressions to match dates and times.

9) Adjuncts extraction: The adjuncts box is the last in the slot priority order. So, after filling the other slots, the remainder entities will be placed into the adjuncts slot.

Before moving to next section, it is important to explain how SentiLecto deals with non canonical constructions, like passive voice and impersonal sentences. All of them are represented using the canonical active voice form.

Ex.: *Juan fue juzgado por el tribunal.*
(Juan was judged by the court.)

El tribunal + juzgar + Juan.
(court + judge + Juan)

Impersonal sentences have an indefinite agent, so, in order to represent them, SentiLecto uses indefinite pronouns *alguien* or *algo,* depending on the subject animacy feature for the verb.

Ex.: *Se reprimió a los manifestantes.*
(Protesters were repressed)

Alguien + reprimir + manifestantes.
(Someone + repress + protesters)

GLOBAL OPERATORS

Global Operators are the algorithmic implementation of linguistic modality. The significance of modality is that we should know when we are in front of a real fact, in order to perform subsequent tasks, like fact mining and sentiment analysis. So, our main interest is to know, in presence of a modality mark, if we should extract a fact or not. Modality could be explained as a set of linguistic resources that allows language users to express what is, what would be, what may be, and what should be.

In addition to Affirmation, SentiLecto can identify and distinguish between other seven types of modality:

- Negation
- Yes/No Question
- Possibility
- Condition
- Imperative
- Future
- Volitive

If SentiLecto identifies the presence of a global operator, the utterance mapped onto the following SVO slots will not be considered as a real fact but an hypothetical situation that didn't happen. So, this will be taken into account in some subsequent tasks, like sentiment analysis.

SentiLecto identifies the use of certain modality through a set of modality marks or indexes, specific for each kind of modality. The search scope of global operators is the clause. However, SentiLecto works with the idea of inherited modality: in certain cases, inside a complex sentence, the subordinated clause could inherit the main clause's modality. This is implemented using a list of *verba dicendi.*

Ex.: *Supongo que Telefónica mejoró sus servicios.*
(I guess Telefónica improved its services)

Main sentence: *yo + suponer* (Possibility)
Subordinate clause: *Telefónica + mejorar + sus servicios.*

As we can see in the previous example, the subordinate clause *“Telefónica mejoró sus servicios”* doesn't refer to a real fact, but there is no modality mark inside the clause. Consequently, in these cases, the subordinate clause has to inherit the main sentence's modality.

This kind of language behaviour would be difficult to solve using a machine learning paradigm because it is difficult to catch it using induction.

Following, we will examine and analyze each kind of modality:

1) Negation: To identify it is easy, negation adverbs always indicate a Negation modality. Some negation verbs, like “dudar”, “refutar” and “desmentir” also point at a Negation modality. And also indefinite pronouns like “nadie” or “ningún”.

The main problem is to identify the Negation scope: we don't know which part of the sentence is being negated. See the examples below:

a) *Telefónica no mejoró su servicio, mejoró su atención al cliente.*
(Telefónica did not improve their service, it improved its customer service)

b) *Yo no trabajo aquí, trabajo en mi casa.*
(I do not work here, I work in my house)

c) *Yo no trabajo aquí, Mariana trabaja aquí.*
(I do not work here, Mariana works here)

In a) negation affects the verb. But in b), it affects the adjunct. And in c), it affects the subject. This, from an algorithmic point of view implies a big problem: What exactly is being negated?

To deal with this trouble, SentiLecto has to postulate a standard scope related with negation. The answer, at least initially, will be that when we find a negation mark, we will not be able to extract a fact with certainty.

2) Yes/No Question: It is easy to identify questions, because they are pointed by special punctuation marks. A first naive and wrong approach to questions, would say that when we find a question, we have to discard it as a fact. But quickly some counterexamples appear:

Ex.: *¿Gracias a quién Telefónica mejoró su servicio?*
(Thanks to whom Telefónica improved its service?)

In the previous example, it is possible to extract a fact. Due to this reason, SentiLecto divides questions in two main groups: partial and total questions. Partial questions, or Yes/No Question (in our terminology) imply a non-real modality, as in the following example:

Ex.: *¿Telefónica mejoró su servicio?*
(¿[Did] Telefónica improve its service?)

In total questions, instead, we can find a fact, as in the first example, where we could extract *Telefónica + mejorar + su servicio.*

3) Possibility: Language has a wide variety of resources to point at Possibilities:

- Verbs in subjunctive mode:

  Ex.: *Probablemente, el servicio de Telefónica mejore el próximo semestre.*
  (Probably the service of Telefónica [will] improve next semester)

- Some adverbs, like "quizás" (*maybe*)

  Ex.: *Quizás venda mi auto.*
  (Maybe I [will] sell my car)

- Some verbs, like "dudar" (*doubt)*

  Ex.: *Los diputados dudan que se apruebe la ley.*
  (Deputies doubt that the law [will] pass)

- Verbs in conditional tense:

  Ex.: *Según ciertas versiones, Telefónica mejoraría sus servicios.*
  (According to some versions, Telefónica would improve its services)

However, conditional has at least two different uses in Spanish: to mark a possibility and to point at a future in the past. How to distinguish? It is a big challenge. Nowadays SentiLecto can't distinguish them. We can also add a third use, a colloquial one, that expresses a factual present meaning: *"No me estaría gustando esto".*

In many cases, the devices that point at a Possibility modality are convergent, for example when we use an adverb like "quizás", we always continue the sentence with a verb in subjunctive mode.

Note that not all the verbs in subjunctive mode imply the absence of a fact. Clauses with *factitive verbs* (like "alegrarse" o "preocuparse", for example) imply a fact, although it is expressed in subjunctive mode.

Ex.: *Me alegra que el servicio de Telefónica haya mejorado.*
(I am glad that Telefónica's service has improved)

Ex.: Me preocupa que el servicio de Telefónica no haya mejorado.
(I worry that Telefónica's service has not improved))

SentiLecto can solve successfully this kind of constructions with factitive verbs, which constitute an exception to the general rule.

4) Volitive: The Volitive modality could be identified through the semantic of a set of verbs, like “querer”, “desear”, “pretender”, “necesitar”. Volitive modality (from Latin *voleo, to wish)* only expresses the act of desiring or the fact of having desired something, but the aim of that desire is not a real fact.

Ex.: *Anhelo que Telefónica haya mejorado su servicio.*
(I hope Telefónica has improved its service)

5) Imperative: The Imperative modality is mainly identified through the verbal imperative mode and through some deontic verbal periphrasis like “tener que” (*have to).* In this deontic verbal periphrasis, although the verb is in indicative mode, we can recognize the semantic value of an order.

Ex.: *Telefónica tiene que mejorar su servicio.*
(Telefónica has to improve its service)

6) Future: There are two main resources to refer a future action, the future tense and the verbal periphrasis *ir + infinitive.*

Ex.: *Juan comprará valiosas obras de arte.*
(Juan will buy valuable works of art)

Ex.: *Mi hermano va a ir a España de vacaciones.*
(My brother is going to go to Spain on holidays)

7) Condition: It is difficult to distinguish between the two senses of the word *si* (In Spanish, *se* in Portuguese)*,* the conditional and the one used to express indirect questions.

Ex.: *Si Telefónica mejora su servicio, es un acontecimiento digno de ser celebrado.*
(If Telefonica improves its service, it is an event worthy of being celebrated)

Ex.: *Si el vendría, yo pregunté.*
(Whether he would come [or not], I asked)

Because of this, currently the conditional modality of the protasis is not inherited by the apodosis.

SENTIMENT ANALYSIS

Sentiment analysis is one of the standard tasks in the Natural Language Processing industry. It consists in extracting subjective information in source materials like news, social media, etc. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic.

Some decades ago, sentiment analysis was based only on detecting and counting positive or negative words. Nowadays, SentiLecto owns a great syntactic knowledge that allows it to build an accurate fact representation and to perform an entity based sentiment analysis. It isn't enough to know whether a text talks positively or negatively about something, we need to know which entities are the receptors of that valuation.

It is advisable to highlight that, in order to perform an accurate sentiment analysis we depend on the appropriate performance of previous processes, like anaphora resolution, identity matching or global operators. Once we have built a good fact representation model and a reference model, we only have to map valuations over the right entities.

SentiLecto mainly uses two resources in order to perform sentiment analysis. One of them is scripts_verbales.csv, which we commented previously. As we said, it has syntactic and semantic information for each verb. In relation with this task, a brief analysis of the verb behaviour related to subjective information shows that each verb spills opinion over elements that have different syntactic functions.

Ex.: *Juan odia a María.*
(Juan hates María)

Ex.: *Juan difamó a María.*
(Juan slandered María)

A brief analysis of the semantic of these verbs will conclude that in the first example, the verb *odiar* spills a negative valuation to the object “María”. However, in the second example, the verb *difamar* also spills a negative valuation, but over the subject “Juan”.

There is a column in scripts_verbales.csv that encodes this kind of information. There are six types of sentiment spilling, marked with capital letters A, O, N, D, S, E. The following table summarizes them:

Types of sentimental spilling		
Code	*Category*	*Description*
A	Agent	The verb spills an opinion over the subject
O	Object	The verb spills an opinion over the object
E	Equative	The verb spills the object valuation over the subject
N	No Spilling	The verb doesn't spill any valuation
D	Double Spilling	The verb spills a valuation over the subject and the object, at the same time
S	Switch	The verb spills a positive valuation over the subject and a negative one over the object, or vice versa, at the same time

On the other hand, there is a resource called sentiwordnet.csv, which lists all the words that have a subjective valuation, either it was positive or negative.

As we said above, global operators affect sentiment analysis mainly because when there is no real fact we have to take into account each modality and modelize how it affects the sentiment spilling. To illustrate it, we can show some examples:

a) *Juan fue encarcelado.*
(Juan was imprisoned)

b) *Juan no fue encarcelado.*
(Juan was not imprisoned)

c) *¿Juan fue encarcelado?*
(¿Was Juan imprisoned?)

d) *Juan tiene que ser encarcelado.*
(Juan has to be imprisoned)

e) *Si Juan fuera encarcelado, iría a visitarlo.*
(If Juan was imprisoned, I would go to visit him)

f) *Juan será encarcelado.*
(Juan will be imprisoned)

g) *Juan sería encarcelado.*
(Juan would be imprisoned)

h) *Deseo que Juan sea encarcelado.*
(I hope John will be imprisoned)

As you may realize, all the examples belong to the same fact (*alguien + encarcelar + Juan,* according to this active form*)* with different modalities. In a) we present an assertion in past tense, we are affirming that Juan was jailed. So, according to the entry for *encarcelar* in sentiwordnet.csv, a -30 score will be spilled over the object (remember that, to perform sentiment analysis we consider the normalized active voice form).

In b) we are negating the same fact. Negation modality affects sentiment analysis in this way: it multiplies the sentiment score by -0.5. Consequently, a negative valuation will be converted in a positive valuation (and vice versa), and the entity *Juan* will receive a positive score of 15.

Cases c), d) and e) belong to Yes/No Question, Imperative and Condition modalities, respectively. In these three cases, modality will stop the sentiment spilling. So, *Juan* will not receive any valuation.

Future (example f) refer to a real fact that did not take place yet. So, the Future modality will not affect sentiment spilling, and *Juan* will receive a valuation of -30, just like in a).
Finally, the last two examples belong to Possibility and Volitive modality. They have the same behaviour: in presence of any of these global operators the score detailed in sentiwordnet.csv will be multiplied by a coefficient of 0.1. So, the valuation of *Juan* will be -3 in these two sentences.

ANAPHORA RESOLUTION

An anaphora is the use of an expression (generally a pronoun) whose interpretation depends upon another expression in a specific context, which is called antecedent.

Ex.: ***Mi hermano*** *me lleva dos años.* ***Él*** *es muy distinto a mí.*
(My brother is two years older than me. He is very different from me)

Ex.: *Juan rompió* ***la pelota****, pero prometió que* ***la*** *iba a arreglar.*
(Juan broke the ball but promises that would fix it)

Anaphors present a challenge to natural language processing and computational linguistics, since the identification of the reference can be difficult. But, once you detect it, this implies a big improvement in tasks like identity matching, fact mining and sentiment analysis.

Traditionally, linguistic theory divides anaphors in two types: endophoric and exophoric. The first group is formed by anaphors that refer to an element that is present in the discourse (as in our examples), while the second group is formed by anaphors whose referent belongs to the outer world. This last group is out of the scope of a NLU engine, so we will focus in the first group. We will find the antecedent for relative pronouns and personal pronouns in third person. That is because pronouns in first and second person tend to refer to exophoric elements. The same happens with demonstrative pronouns, they mostly belong to exophoric elements.

As you may know, pronoun form varies according to the syntactic function it takes up in the sentence, we know this behaviour as case inflection. For example, the third person masculine pronoun is “él” for the nominative case, “le” for the dative case, “lo” for the accusative case, and “su” for the genitive case.

When we find a pronoun, we have to find the syntactic function that it executes (based mainly in its case) and its referent. It will be searched at different levels, first in parent clauses, then in previous clauses, and finally in previous sentences.

To find the referent for personal pronouns in nominative case, previously we have to have reinstate the tacit subject. As we said before, we will solve nominative personal pronouns in third person: *él, ella, ellos, ellas.* We will not solve the referent to the neutral pronoun *ello.* Obviously, a pronoun always has to share the morphosyntactic information (MSI information) with its referent.

These personal pronouns referents will be searched in a certain order, according to the sentence voice. In active voice: subject, DO, IO, regimen, equative, instrument/goal and adjuncts. And in passive voice: DO, subject, instrument/goal, adjuncts.

Regarding accusative personal pronoun, we will solve all of them: *lo, los, la, las.* We will find their referent in these previous syntactic functions: DO, subject, regimen, IO, equative, instrument/goal, adjuncts, in active voice sentences. And subject, DO, instrument/goal, adjuncts, in passive voice sentences.

Before talk about relative pronouns, it is a good idea to point the difference between subordinate sentences and relative sentences. Although both could be headed by “que”, their values will not be the same in the two cases. See this examples:

Ex.: *El hombre vociferó* ***que*** *el mundo se acabaría.*
(The man shouted that the world would end)

Ex.: *El hombre* ***que*** *murió era mi padre.*
(The man who died was my father)

In the first example, the word “que” doesn't refer to any entity; we usually say that it is empty. However, in the second example, the word “que” works as a pronoun, and refer to “El hombre”. So, only when we find a relative sentence we have to find a referent to the header.

Relative pronouns could be solved at sentence or clause level. There are three functions that cover three possible cases:

a) Possessive relatives: “cuyo”, “cuyos”, “cuya”, “cuyas”
b) Relatives without any article or preposition: *que, quien, quienes,* except *donde* and *adonde,* that will not be solved.
c) Relatives with an article or preposition: *el que, los que, la que, las que, del que, por lo que, el cual, los cuales, etc.* Except the neutral form *lo cual.*

The relative could be a possessive relative, like “cuyo”, “cuyos”, “cuya”, “cuyas”. In this case, we have to find the possessed entity and the owner entity. The possessed entity will be the first entity on the right side of the relative. And the owner is the referent of the possessive relative. It will be destined to a slot according to the sentence voice: if it is active, it will fill the subject slot, if it is passive, it will fill the direct object slot.

Relatives without articles or prepositions will be solved basing on the same principle, the destination will be assigned according to the sentence voice: if the sentence is active, the relative will fill the subject slot, and if it is passive, the relative will fill the direct object slot. The referent will be the closest entity on the left side that share the same morphosyntactic features, prioritizing top levels.

To solve relatives that come with a preposition or with an article, first of all we have to evaluate the gender and number requirements of the relative, which are provided generally by the article. Then, we evaluate the possible structures that it will fill. If there is a preposition before the relative, it could fill the regimen or the adjuncts slot. If the preposition is one of the set of valid instrument/goal prepositions, it also could fill the instrument/goal slot. If the preposition is "a", it could fill the direct object slot. Otherwise, when there is not preposition, the relative could only fill the subject or the direct object slot.

All this possible structures will be analyzed one by one, in the same order they were presented. If The verb requires any animacy feature for this slot, it will be also taken into account. There are two methods that do this work, one for active sentences and other for passive sentences.

IDENTITY MATCHING

As anaphora resolution, identity matching has a primary importance for the appropriate execution of subsequent tasks. As we know, names have a structure and typology that are influenced by many cultural and regional factors. Us humans know these differences and usually recognize them without even thinking twice. Machines have to be taught how to make these distinctions. That's the more complicated bit.

The identity matching algorithm will depend on the type of the named entity. Of course, when an entity appears identically twice, it will always match.

SentiLecto modelizes the structure of a person's name in order to catch the whole name and consequently to improve the subsequent identity matching. In this modelization, a name is formed by:

- Honorific, like "Dr.", "PhD". Of course, it is an optional item.
- Name: the first capitalized word.
- Middle names.
- Connective particles (like "*della*" in "*Fernando della Maggioria*")
- Last name: the last capitalized word.

Persons will match when they have the same name and/or the same last name. We illustrate it in this table:

First Occurrence	Second Occurrence	
Martín	Martín Pérez	MATCH
Martín Pérez	Martín	MATCH
Martín Gómez	Martín	MATCH
Martín	Martín Gómez	MATCH

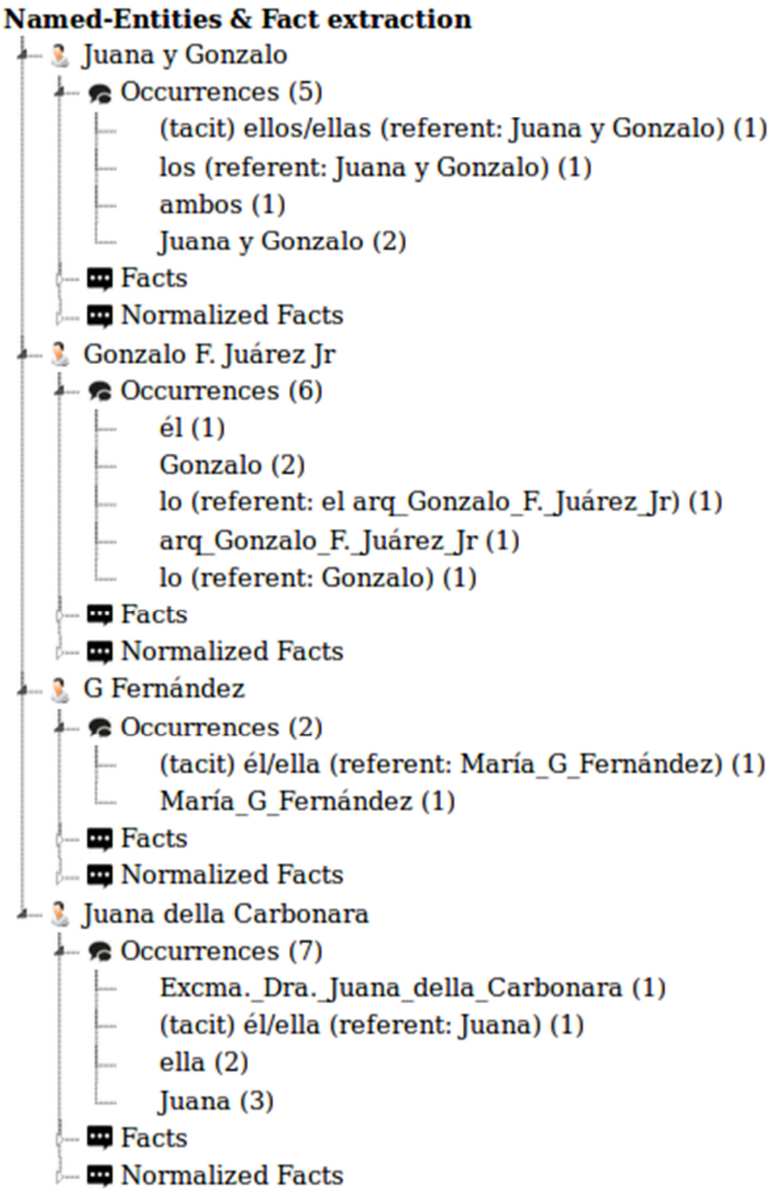
The other types of named entities (PLACES, ORG and MISC) will match only if their core is identical, with the same semantic classification.

In the example below, we show a text with several references to two persons and all its occurrences, and the SentiLecto output.

> *Gentil y amable como pocas, la Excma. Dra. Juana della Carbonara saludó al Arq. Gonzalo F. Juárez Jr. en las oficinas de la gobernación,*

> *ante la mirada del propio gobernador, dejándolo perplejo. María G Fernández se cruzó con ella y le preguntó por él. Juana y Gonzalo parecen una pareja disfuncional cuando discuten, no? Aunque los admiro a ambos por igual, supuse que ella lo iba a hostigar. No obstante, destacó las buenas intenciones de Juana, en favor de la cual debo decir que los flirteos mutuos han sido mitigados por el tiempo. ¿Quién les robó sus sueños y deseos? ¿Desaparecieron para siempre? Si la vida se propone separar a las personas que aprecio, me resisto a asumir responsabilidades. En tal caso quiero permanecer como un eterno niño, resistiendo a la embestida del tiempo, aunque reconozco que resulta inconducente que adoptemos esta postura. Juana y Gonzalo desperdiciaron su oportunidad de ser felices. Yo no voy a tirar la toalla así nomás.*

In the next image you can see how the identity matching and the anaphora resolution algorithm work together to identify and group all the references to a same person.



Named-Entities & Fact extraction

- Juana y Gonzalo
  - Occurrences (5)
    - (tacit) ellos/ellas (referent: Juana y Gonzalo) (1)
    - los (referent: Juana y Gonzalo) (1)
    - ambos (1)
    - Juana y Gonzalo (2)
  - Facts
  - Normalized Facts
- Gonzalo F. Juárez Jr
  - Occurrences (6)
    - él (1)
    - Gonzalo (2)
    - lo (referent: el arq_Gonzalo_F._Juárez_Jr) (1)
    - arq_Gonzalo_F._Juárez_Jr (1)
    - lo (referent: Gonzalo) (1)
  - Facts
  - Normalized Facts
- G Fernández
  - Occurrences (2)
    - (tacit) él/ella (referent: María_G_Fernández) (1)
    - María_G_Fernández (1)
  - Facts
  - Normalized Facts
- Juana della Carbonara
  - Occurrences (7)
    - Excma._Dra._Juana_della_Carbonara (1)
    - (tacit) él/ella (referent: Juana) (1)
    - ella (2)
    - Juana (3)
  - Facts
  - Normalized Facts