Natural Language Processing for a Social Network Analysis

supervisor
prof. Amato Flora

candidate
Romano Andrea
matr. N46002467

March, 2019
Contents

1 NLP-Natural Language Processing 2
   1.1 IE-Information Extraction ......................................... 4
   1.2 Limiti dell’NLP ......................................................... 4

2 SNA-Social Network Analysis 7
   2.1 Identify and study the social network .......................... 8
   2.2 Choice of data ......................................................... 9
   2.3 Access the data ....................................................... 10
   2.4 Social Network log Analysis ...................................... 11

3 Tools 12
   3.1 API Facebook .......................................................... 12
   3.2 Spring Social App ..................................................... 14
       3.2.1 Accessing Facebook Data ..................................... 14
   3.3 GATE ................................................................. 19
       3.3.1 Generation of corpora ........................................ 20
       3.3.2 ANNIE – A Nearly New Information Extraction System .. 21

4 Facebook-Fetch-Data 25

5 Results of the experiment 31
   5.1 Catch post from Facebook ......................................... 31
   5.2 Semantic analysis of the posts .................................... 33
Natural Language Processing for a Social Network Analysis

Introduction

In recent years, many companies have shown an increasing interest to use Big Data and NLP technology to create added value. The first challenge we face when we talk about Big Data is that these are almost always unstructured, because written in natural language. This limits the understanding of data by machines. Examples of unstructured data are messages, e-mails, blog articles, social feeds, etc. The semantic information contained in the data, poses a few challenges to the NLP (Natural Language Processing), a branch between computer science and linguistics that studies the interactions between computers and human language. These functions can be very useful especially for marketing purposes, for managing the customer experience, for statistical surveys, etc. These analyzes allow us to understand what users are talking about on various social media, allowing us to identify new trends and interpretations of public opinion. In this thesis we will focus on the development of a web application that retrieves and saves, after authorization to process data, some user data on Facebook. This data will be processed by the NLP GATE software in order to extract from the user’s posts, through a series of jape rules, the latter’s opinions on a specific topic. To achieve our goal we will see, in order, the following topics that will give us the opportunity to develop our application first and, subsequently, the analysis of the results obtained through the application.
Chapter 1

NLP-Natural Language Processing

Natural Language Processing refers to the computer processing of natural language, for any purpose, independent of the level of depth of the analysis. Natural language means the language we use in everyday life and is synonymous with human language, mainly to be able to distinguish it from the formal language, including the language of computers. As it is, natural language is the most natural and most common form of human communication, and not only in its spoken version, even the written version is growing exponentially in recent years, since mobile internet has grown with new social media. Compared to formal language, natural language is much more complex, often containing ambiguity, which makes it very difficult to elaborate. In greater detail, the NLP provides solutions to analyze the syntactic structure of the text, associating the respective morphological categories (eg name, verb, adjective) to individual words, identifying entities and classifying them in predefined categories (eg person, date, place), extracting syntactic dependencies and semantic relationships. We can therefore schematize a complete processing process with the following analyzes:

- Lexical analysis: that is, breaking down a linguistic expression into atomic elements called tokens; examples of tokens are words, punctuation marks, numbers, etc.
• Grammatical analysis of the text.

• Syntactic analysis: a sentence is organized hierarchically in constituents, the syntactic analysis is, therefore, the analysis that allows us to identify this organization.

• Semantic analysis: assignment of a meaning to the syntactic structure we have seen previously.

The history of this field of research is usually started in the fifties, when Alan Turing published his article "Machine and Intelligence", in which he proposed his famous test to assess the ability of a computer to show intelligent behavior, indistinguishable from those of a human being, conversing in natural language. Until the eighties, most of the NLP approaches tried to formalize the vocabularies and rules of natural language (rule based approach) in computers, but with results that were not always satisfactory. The rigidity and non-scalability of handwritten rules have not, in fact, made possible the extensive management of extreme variability and ambiguity, functional, lexical and syntactic, of natural language. At the beginning of the eighties and, more extensively in the nineties, the first approaches to statistical or corpus-based NLP appeared. In particular, they have tried to solve the problems of rules-based approaches, independently learning the lexical and structural preferences from large corpora, without requiring a preliminary manual coding (approach with Machine Learning). Most of the NLP statistical approaches, still widely used in different commercial products, elaborate the set of words present in a text (bag-of-words), sometimes enriched with the different morphological categories, allowing a better management of the different types of language ambiguity and, at the same time, guaranteeing robustness, generalization and resilience in the presence of errors. However, given the complexity of the analysis process and the possible vastness of the application domain, machine learning techniques are often combined with those of the rule based type.
1.1 IE-Information Extraction

The automatic extraction of structured information from unstructured or semi-structured documents is called Information Extraction (IE). In an IE process there are three levels on which to intervene:

- **Named Entity Recognition**: this is the simplest task that is limited to identifying the entities in a sentence.
- **Coreference Resolution**: detection of coreference links, the set of tokens that refer to the same referent, and of anaphora, references between parts of the text that are more or less distant from each other.
- **Relationship Extraction**: recognition of associative links between entities.

When making an IE often the three levels described above must be performed simultaneously, for example in the phrase "Mario Rossi resides in Rome. But he was born in Milan." we must: identify the entities in the sentence (Named Entity Recognition); understand that "he" refers to "Mario Rossi" (Coreference Resolution); understand that Rome is his residence and that Milan is his birthplace (Relationship Extraction).

1.2 Limiti dell’NLP

Despite the convincing results obtained in different applications, for example with search engines and knowledge extraction, there are still different open research challenges, which can be summarized as follows.

- **Automatic processing of natural language for scenarios or application scenarios with limited availability of annotated data.** In-depth learning approaches, for example for controlled class, training data with the classes or phenomena they face (the topic or polarity) are required. Annotating the dataset...
is an intensive process that requires time and human resources to complete it. As for the languages used, such as English, or for widespread application scenarios, such as sentiment analysis, it is a wider availability of annotated data and therefore, a greater possibility of having profound learning models available, for many others or scenarios there is a lack of data that can be used for this purpose. Currently, different approaches are being studied to deal with a problem, from the definition of a "universal" model of language starting from common aspects identified between different languages, to the transfer of styles and models or a different task, up to non training strategies, supervised or supervised with small annotated data sets.

- The "real" understanding of natural language in an automatic way, that is the ability to read and understand a text in a similar way to a human. From a more specifically cognitive perspective, the understanding of natural language in human does not occur in isolation, without any information on the context and the surrounding environment, but is closely related to the perceptive experience with the external world. Reproducing in the machines the comprehension and production abilities of the human’s language cannot do without creating and correlating conceptual representations of objects from the surrounding environment. To this end, an increasing number of approaches are emerging that integrate knowledge from cognitive sciences and neurosciences to create conversational systems capable of simulating high-level cognitive functions underlying the development of infantile language in the human brain, learning through experience and interaction with the external environment.

- The ability to carry out common sense reasoning to understand language in order to resolve ambiguities due to implicit or unspecified knowledge. For common sense reasoning we mean, for example, those that allow us to un-
nderstand who a pronoun refers to in a sentence or that the penguin does not fly, despite being a bird and birds flying. To date, despite several advances made, teaching a machine to make common sense reasoning still remains an unresolved problem. The absence of common sense in machines represents one of the barriers that have limited the ability both to understand the phenomena that occur in the surrounding world and to communicate naturally with human, not allowing them to behave in a reasonable manner in unknown situations and to learn from new experiences not covered in the training phase. Recently, ad hoc datasets have been created and international research programs have been proposed specifically aimed to providing solutions to this challenge, requiring the integration of knowledge coming from heterogeneous sectors, from cognitive psychology, to linguistics up to computer science.

In conclusion, the key to addressing these challenges lies, in part, in developing advanced solutions that take inspiration from different disciplines, from neuroscience to cognitive psychology, to linguistics and computer science, to represent, understand and generate text or speech in language natural in a similar way to the human, and on the other hand, in having large amounts of data available, even aligned between different languages, to train and validate their performance.
Chapter 2

SNA-Social Network Analysis

SNA is the analysis of social networks, that is a methodology of study of relationships that finds application in different sciences: sociology, psychology, economics, anthropology, but also genetics and computer science - areas where the term "social" is understood in a broad sense, no longer with direct reference to the relations between human beings, but more generally to the relations between individuals, entities, actors of a relationship. The analysis of social networks makes it possible to understand the connections in a group of people, their links and the intensity of these links; identify the most influential people in a group, find out which people are the key to connecting two individuals or two groups of individuals who would otherwise not communicate with each other. In addition to understanding how a social network is structured, it is extremely interesting to understand what this network can tell us, or the data that this network generates, and that can be exploited for marketing activities, statistics, etc. SNA and Social Networks are a perfect combination, with Social Networks we have access to millions of data generated, even unknowingly, by millions of users and all at almost no cost. Through the data generated by a group of users it is possible, for example, to carry out a market survey on a given product, improve it by responding to the common opinion of this group and, perhaps, have it publicized by the most influential users of
the network analyzed. When it comes to SNA there are three problems to deal with:

- Identify and study the restricted group of users who can provide us with the information we need.
- Isolate, among many others, the user data we need.
- Physically access user data through its authorization.

Let’s look at each problem in detail.

### 2.1 Identify and study the social network

The easiest way to study a social network is to have a structured view of it. The best way to structure a social network is through the use of graphs: each node of the graph represents a single user, the arcs between one user and another represent the interactions between them. These graphs can be traced by hand, identifying the links between the various users, however this approach is outdated, both due to the growing complexity of these networks, and to the creation of free tools made available by the same social networks for the creation of these graphs. Let’s see, with a very simple but effective example, how a social network structured with graphs looks like:
Looking at the graph it is immediate to understand that the user "Bob" is connected to all the other users of the network, Bob seems to represent the ideal user to recruit to advertise a possible product in the event that our social network analysis has been carried out for marketing purposes.

2.2 Choice of data

The data obtained from a user can be of various nature and, in particular, we can identify the five most important:

- Textual data: comments, posts ...
- Data related to the analyzed network: Facebook friends, Twitter or Instagram following system ...
- Actions: sharing, reactions, "like" ...
• Hyperlinks

• Useful data for SEO purposes (Search Engine Optimization)

The choice of data types to use depends on the purpose for which we are performing this social network analysis. The types of data that can be taken can, of course, be combined together to achieve more complex objectives. The Cambridge Analytica case, a scandal involving the latest US elections, is a perfect example that will give us interesting insights also for the following subparagraph. The Cambridge Analytica company carried out social network analysis by exploiting the data of American citizens on Facebook to understand its political orientation and try to convey its thinking with targeted advertising. To understand the political thought of a user, different types of data have been taken, in particular: textual data (posts and comments), likes and people of inspiration for the user.

2.3 Access the data

As regulated by the GRDP - General Regulation on Data Protection, in order to process the data of a user, the latter must be aware of all his data that we intend to use and consent to their processing. Hardly a user will be willing to share their data to have them analyzed, a now widespread technique of social engineering is to offer a reward to the user in exchange for access to his data. The most common reward is the use of your social account to register to a website or a mobile application, the user will have the convenience of registering quickly avoiding the classic registration phase with data compilation and we will have access to user data generated on that social network. Returning to Cambridge Analytica, the scandal came about precisely because of a GRDP violation; the data of the users involved were prewashed data via the "Thisisyourdigitallife" application, an application that provided a series of tests and games to users, an
intellectual property application of a psychology professor Aleksandr Kogan. By accessing the app, the user gave consent to the processing of his data in a manner consistent with the GRDP, but these data were sold by Kogan to the Cambridge Analytica company, thus breaking a Facebook regulation and, in general, a GRDP regulation that claims to be able to transfer user data to third parties only if the latter consents to this clause.

2.4 Social Network log Analysis

The Social Network log Analysis is a particular SNA, it doesn’t focus on the study of a whole social network, it deals only with the data to be taken from a social account. The problems of data selection and access to data remain present, but we don’t worry about defining and studying the social network. Another reason why it differs from the SNA is that, while the latter is carried out also on non-virtual social networks, the Social Network log Analysis is based only on data that can be taken from virtual social networks, so we can take advantage of all the tools provided by the various social networks to make our work even easier. In this thesis work the Social Network log Analysis was used, the goal is to collect the data of a user from his social account and analyze that data independently of the user’s relationship with the rest of his social network.
Chapter 3

Tools

In this chapter we will analyze the tools we will use to create, first, the web application from which the user will allow us to take their data and, subsequently, the software that will allow us to analyze the data.

3.1 API Facebook

Facebook provides an API to manage access to our application and, possibly, to process the data that the user agrees to provide. The API made available by Facebook is API Graph, to use API Graph in our application it’s necessary to access the site https://developers.facebook.com with a Facebook account that will be a reference for our application, here we must declare: purposes, platform (Web, smartphone app, etc.), intellectual owners. Once the registration procedure is completed, Facebook will provide us with a unique ID and a secret key that will identify our application, ID and secret key will be included in the application code.

<table>
<thead>
<tr>
<th>ID app</th>
<th>Chiave segreta</th>
</tr>
</thead>
<tbody>
<tr>
<td>255833052022392</td>
<td>●●●●●●●●●●</td>
</tr>
</tbody>
</table>
intend to use. We must evaluate what data are of interest to us and ask the user for permission to be able to withdraw and use them, even in this case Facebook comes to us with an exploration tool from the Graph API. Just connect to the site https://developers.facebook.com/tools/explorer/ to receive a token, this token is a string of characters to insert in our application and which will then be analyzed to extract the list of data for which we request authorization for processing. To receive the token with the data of our interest, just click on "receive token" -> "receive user token" and select from the possible data.

These are the data that Facebook allows to withdraw when this thesis was written (03/2019), the types of data that can be taken vary to adapt to changes in the GRDP and to Facebook updates. For example, before the Cambridge Analytica scandal, which we discussed in Chapter 2, it was possible to take a list of the user-inspired people, this was removed because it was one of the types of data used by Cambridge Analytica to understand 'political orientation of American voters and try to convey their thinking through targeted advertising.
3.2 Spring Social App

Spring social is an open source framework that allows you to connect applications to the API of providers like Facebook, Linkedin and Twitter. In our case we will use Spring Social Facebook in a Java application. By connecting to the site https://projects.spring.io/spring-social-facebook/ it’s possible to consult the guide and also download the skeleton from which to develop our application, in particular we will use the "Accessing Facebook Data" skeleton. Before importing Accessing Facebook Data we must verify that our development environment allows us to develop spring applications, in this case we will use Eclipse (version 2018-12). Basic Eclipse does not have the option to develop spring applications, but just install the following tool: http://marketplace.eclipse.org/content/spring-tools-3-add-aka-spring-tool-suite-3.

3.2.1 Accessing Facebook Data

After installing the spring tool we can import Accessing Facebook Data as an existing Maven project, the main objective of Maven is to help the developer manage the status of a project, and Maven also allows: facilitating build processes, providing information quality of the project, to provide guidelines on the best practices being developed; all information contained in a pom.xml file. The pom is an XML file and guides the execution in Maven clearly defining the identity and structure of a project in all its aspects, everything is described in the POM: general information of the project, dependencies, compilation process and secondary phases as the generation of documentation. Let’s see the pom.xml:
Natural Language Processing for a Social Network Analysis
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.springframework.social</groupId>
  <artifactId>spring-social-facebook</artifactId>
</dependency>

</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
    </plugin>
  </plugins>
</build>
Accessing Facebook Data provides us with 3 other fundamental files: application.properties, SpringFacebookExampleApplication.java, FacebookController.java; let’s see them one by one.

- **application.properties**: This .properties is the file in which we will have to enter the ID and secret key of the registered application on https://developers.facebook.com (as we saw in the Facebook API chapter).

```
1 spring.social.facebook.app-id=
2 spring.social.facebook.app-secret=
```

- **SpringFacebookExampleApplication.java**: This .java is the launch point of
our application, it will contain the public static void main (String [] args), we also use the SpringApplication.run (Class, String []) which is used to start an spring application.

```
@SpringBootApplication
public class SpringFacebookExampleApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringFacebookExampleApplication.class, 
        args);
    }
}
```

- FacebookController.java: This is the heart of our application, here we deal with:

  1. Connect to Facebook through the connectionRepository class

  2. Request authorization to process data using the token we have already discussed in the Facebook API chapter

  3. Take and group user data of interest to us via facebook.fetchObject ("me", User.class, fields);

```
@Controller
@RequestMapping("/")
public class FacebookController {

    private Facebook facebook;

    private ConnectionRepository connectionRepository;
```
GATE is an open source free software that includes an integrated development environment (IDE - Integrated Development Environment) for the use of NLP components already present in the application and a set of useful plugins to install if necessary. In particular it provides an IE system, called ANNIE, which includes a series of components normally used in a typical NLP processing and which have proved particularly useful. Version 8.5.1 of GATE was used for this test.
Before describing the modules used and the procedure implemented, the following concepts are defined on which GATE is based, necessary for understanding what is described below:

- Applications Resources (AR): allow you to define sequences of processing resources that can process individual documents or the entire corpus.

- Language Resources (LR): allow you to view the document or corpus you are working on.

- Processing Resources (PR): display the modules activated by the user that can be selected and inserted in the sequences mentioned at point one.

- Datastores (DS): makes the corpus and annotations created by the user on documents persistent by storing them in a datastore.

- Document: textual element being processed.

- Corpus or corpora: set of documents that will be processed uniformly by the various modules selected by the user.

- Annotation: annotation that will be created by analyzing the document.

- Annotation set: group of related annotations.

### 3.3.1 Generation of corpora

First of all it is necessary to load the documents to be processed: selecting LR, uploading a document (in our case the user posts), creating a corpus containing that user and finally populating the corpus adding all the other users at our disposal. In this way we obtain a set of documents managed as if they were a single large document.
To ensure that the annotations created on the documents remain "persistent", that is memorized, it is necessary to define a datastore: of the two possible types, a SerialDataStore based on a Java serialization system was chosen. At this point you can proceed with the creation of the annotations.

### 3.3.2 ANNIE – A Nearly New Information Extraction System

GATE includes a complete Information Extraction system called ANNIE (a Nearly-New Information Extraction System). This system is not automatically loaded by the program but must be selected by the user as a plugin, configured and possibly set for automatic loading at each GATE start. It includes a series of modules briefly illustrated below.

- **Document Reset** This resource allows you to reset the annotations added to the document in the various elaborations and is particularly useful when performing repeated processing on the same corpus to test the different results;
any original notes remain unchanged.

- **Tokeniser** The purpose of this resource is to divide the text into "pieces" such as numbers, punctuation and words. As output, SpaceToken or Token annotations are obtained, with attributes like string, orth, category, length, etc. associated by default.

- **Sentence splitter** The PR Annie Sentence Splitter divides the processed document into segments corresponding to sentences.
Part Of Speech Tagger The effect of this PR on the text considered consists in the assignment to each Token of an additional category attribute valued based on a predefined coding whose complete list is available on the GATE user guide.

Gazetteer The role of this PR is to identify entities in the text based on specific lists. This phase, from a practical point of view, does not present particular difficulties; however it requires considerable accuracy in preparing the lists if satisfactory results are to be obtained.

JAPE Transducer JAPE (Java Annotation Patterns Engine) is a pattern-matching language, a mode that allows you to write some rules based on a precise syntax, which will then be translated into JAVA code and executed according to a precise sequence, producing additional annotations as output, changes to existing ones or modification / creation of attributes. A JAPE grammar consists of a .jape file in which a set of phases is specified.
according to a precise logical order. The sequence of phases is justified by the need to create temporary annotations that are transformed into input in the subsequent phases to be combined or transformed.
Chapter 4

Facebook-Fetch-Data

By modifying the Accessing Data Facebook skeleton we get our final application (Facebook-Fetch-Data), let’s see how the code was modified. The application.properties and the SpringFacebookExampleApplication.java do not change, the file to be modified to achieve our goals is FacebookController.java, we must:

1. Insert the permission token obtained from https://developers.facebook.com/tools/explorer/
2. Add the fields of our interest (firstname, name, email, birthday, gender, agerange).
3. Pick up user posts by entering them in a PagedList.
4. Create a text file containing the user’s posts on which to carry out, later, the semantic analysis.
@Controller
@RequestMapping("/")
public class FacebookController {

    private Facebook facebook;

    private ConnectionRepository connectionRepository;

    public FacebookController(Facebook facebook, ConnectionRepository connectionRepository) {
        this.facebook = facebook;
        this.connectionRepository = connectionRepository;
    }

    @GetMapping
    public String getFacebookFeeds(Model model) {
        if (connectionRepository.findPrimaryConnection(Facebook.class) != null) {
            return "redirect:/connect/facebook";
        }
    }
}
Facebook-Fetch-Data will be a web application so we create three html pages:
• facebookConnect.html which allows us to connect to Facebook Login.

```html
<html>
<head>
    <title>Hello Facebook</title>
</head>
<body>
    <h3>Connect to Facebook</h3>

    <form action="/connect/facebook" method="POST">
        <input type="hidden" name="scope" value="user_posts" />
        <div class="forminfo">
            <p>You aren't connected to Facebook yet. Click the button to connect this application with your Facebook account.</p>
        </div>
        <p><button type="submit">Connect to Facebook</button></p>
    </form>
</body>
</html>
```

• facebookConnected.html which will be the page where we will be redirected if the login on Facebook is successful, from this page it will also be possible to access the third and last HTML page.

```html
<html>
<head>
    <title>Hello Facebook</title>
</head>
<body>
    <h3>Connected to Facebook</h3>

    <p>You are now connected to your Facebook account. Click <a href="/">here</a> to see some entries from your Facebook feed.</p>
</body>
</html>
```

• profile.html, here the user can view some of his Facebook profile information.
**Natural Language Processing for a Social Network Analysis**

```html
<html>
<head>
<meta charset="ISO-8859-1"/>
<title>FaceBook-Connect</title>
</head>
<body>

<h1>Hello, <span th:text="${userProfile.name}"/></h1>

<table>
<tr>
<td>First Name: </td>
<td><span th:text="${userProfile.firstName}"/></td>
</tr>
<tr>
<td>Gender: </td>
<td><span th:text="${userProfile.gender}"/></td>
</tr>
<tr>
<td>Email: </td>
<td><span th:text="${userProfile.email}"/></td>
</tr>
<tr>
<td>Birthday: </td>
<td><span th:text="${userProfile.birthday}"/></td>
</tr>
<tr>
<td>Age: </td>
<td><span th:text="${userProfile.ageRange}"/></td>
</tr>
```


Natural Language Processing for a Social Network Analysis
Chapter 5

Results of the experiment

We use all the tools we have studied to achieve our goal: collect a user’s Facebook posts, analyze their syntax with ANNIE’s resources and add a JAPE rule to highlight the topic of interest to us, the TAV.

5.1 Catch post from Facebook

The web application, used to access user data, is a test application and not subjected to Facebook’s security review. We will not be able to use real accounts, so let’s take advantage of creating a test user made available by Facebook to test applications.
We enrich this profile with a series of posts on various topics, some of these concerning political thoughts on TAV.

Let’s now connect to http://localhost:8080/connect/facebook, we will find the following html page:

**Connect to Facebook**

You aren’t connected to Facebook yet. Click the button to connect this application with your Facebook account.

Once this is done, we will be directed to the Facebook login, once we have entered the login credentials we will have to accept the conditions for data processing by our application:
Once you click Ok we will be redirected to our page in localhost with the following screen that gives us confirmation of the successful login:

**Connected to Facebook**

You are now connected to your Facebook account. Click [here](#) to see some entries from your Facebook feed.

Already with the access is generated the .txt containing the posts, by clicking on here the user can access some data taken from his profile, the latter what is not needed for our goal, was inserted only to simulate a real service for the user. What interests us is the content of the generated file "Open Graph User.txt":

This text file is the output sought for this first phase, let’s move on to the second phase, the semantic analysis of the posts.

### 5.2 Semantic analysis of the posts

We import both ANNIE and the newly generated .txt file on GATE:
We need to start processing ANNIE on the corpus containing the newly imported .txt file:

Let’s see the results of the semantic analysis:
Il destino non è un percorso già deciso per te, il destino è un'occasione per fare la scelta giusta.

Non si smette mai di imparare

Un cavallo di battaglia della TAV è sulla facilità di trasporto delle merci. Tuttavia è facilmente reperibile se qualcuno analisi ufficiale di traffici un dato che il Governo dimentica. I valichi alpinisti italiani sono stati utilizzati sotto la soglia del 40%, eccetto il Brennero. Ciò accade perché il flusso di merci prevalentemente utilizzati gli assi di collegamento Nord Sud, ovvero dai porti italiani verso il centro Europa e viceversa, ma il 12% dei traffici italiani si dirigono verso Est, mentre meno del 20% prende la direzione Ovest.

I collegamenti internazionali moderni sono ormai di tipo immateriale (internet) realizzabili in tempi brevissimi e modulabili a costi irrisori. Perseverare con opere colossali legate al cemento e tendono come la TAV è un modo vetusto di guardare al mondo ed al nostro Paese, collegato ai gruppi di potere che portano avanti tecnologie ed interessi altrettanto vecchi.

Informativi sui programmi primi di votare, non scegliete in base alla simpatia/antipatia verso una persona.

Quelli che manifestano contro la TAV hanno tutto il mio appoggio!

Un'infrastruttura da 9,6 miliardi per creare un collegamento che già esiste?

L'unione europea finanzerebbe il 40%, è vero, ma si parla comunemente di un

<table>
<thead>
<tr>
<th>Type</th>
<th>Set</th>
<th>Start</th>
<th>End</th>
<th>Id</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token</td>
<td>0 4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>{category='NNP', kind='word', length=4, orth='upperInit'}</td>
</tr>
<tr>
<td>Token</td>
<td>4 5</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>{category='.', kind='punctuation', length=1, string='.'}</td>
</tr>
<tr>
<td>Token</td>
<td>6 8</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>{category='FW', kind='word', length=2, orth='upperInit'}</td>
</tr>
<tr>
<td>Token</td>
<td>9 16</td>
<td>6</td>
<td>16</td>
<td>6</td>
<td>{category='NN', kind='word', length=7, orth='lowercase'}</td>
</tr>
</tbody>
</table>

13 Annotations (0 selected) Select:
GATE, as in general the semantic analysis through software, has obvious limits, limits given by the language and by the lack of knowledge of some acronyms / terms. We see an example of an incorrect label and an example of terms unknown to GATE:
The latter imagines underline that the acronym TAV is not recognized, we
want to recognize it and also be able to select it quickly since it is precisely our case study. To do this we need to implement a custom JAPE rule, in particular:

**Phase:** Acronym  
**Input:** Token  
**Options:** control = appelt

**Rule:** Acronym1

```
(Token.string == "TAV")
```

We import the .jape file via the PR "JAPE transducer"

We perform ANNIE again, taking care that the JAPE rule comes, in order, after the Tokenizer phase since the Tokens represent the input of our rule; finally
see the results.