

Feedback System Using Sentiment Analysis

Abdulrahman Almonajed¹, Dino Kečo¹,

¹International Burch University, Sarajevo, Bosnia and Herzegovina

abdulrahman.almonajed@stu.ibu.edu.ba

dino.keco@ibu.edu.ba

Abstract – Today, when looking at the quality of an online item, the feedback itself plays a very important role. Based on the feedback we can decide whether the desired item is good or not, get a picture of the seller and so on. Many companies that have online shops display the most positive feedback while hiding bad ones or display only a few of them. In this research, we will help people by automating the process of deciding whether a feedback is positive or negative, which will give them time for other jobs and save money for hiring people who will work on the feedback. Since feedback on online articles is very important today, the process of determining positive and negative feedback should be made as quick and easy as possible. In this research, we will show a very simple and fast way to classify feedback as positive or negative, which means that the main question of this research is how to facilitate and speed up the process of determining the polarity of the feedback. We will use NLP using Python's library called TextBlob. The used algorithm is called Naïve Bayes, it gave the accuracy of around 80%.

Keywords - *feedback, online article, sentiment analysis*

1. Introduction

These days, the number of online stores is growing very fast [1]. We can see that today we can buy whatever we want online. Also, through online shopping we can save a lot of money by being able to find things much cheaper than they are in local stores. By shopping through online shops, we can "escape" arrogant sellers, as well as annoying sellers who follow us during the shopping and "force" us to buy their products. Also, we can save a lot of time by avoiding traffic jams, waiting in line at the store, saving money by not paying for parking, saving our fuel, etc. We can even buy things we don't have in our city or country. For leading companies such as Amazon, Alibaba, eBay, and so on, feedback from every user is very important. They receive thousands of feedback a day, which is very difficult to read and analyze, which is why they need to automate the process. Understanding and analyzing the feedback can improve the user experience, improve the products, and so on, but can also help the online shop owners to know which seller is not doing their job properly, whether it is cheating, etc. Also, there are online applications where we can book an apartment, rent a car, etc., such as on our BTT (Balkan Tourist Travel) application. This kind of web application is now well known in our region, so we decided to create one to facilitate the tourism process in Bosnia and Herzegovina. The application is intended for tourists who visit our country in large numbers. BTT application will make it easier for them to book everything they need during their stay in our country with a few clicks. The main goal of the application is to avoid numerous calls and

misunderstandings between our people and tourists. On the BTT we value feedback, so users can leave feedback on everything that they have used on our application. By doing so, we give our customers the opportunity to express their opinions, which will help us to achieve the best possible service. In this research, we will use the BTT application to apply and test our classification method. For the best process of development, we will be using only one part of the BTT web application. We will perform all the tests and modifications to achieve the best possible results. And if the results are satisfying we will include all the other parts of the application.

Customers' opinion is not only important to large companies it is also important to small companies that are just getting started [2]. Therefore, determining whether the opinion is positive or negative must be automated as soon as possible and in the best possible way. This research will solve this problem and determine whether customers' opinion is positive or negative in a very quick and easy way.

The biggest problem this research solves is the hard work of reading the opinions, which can be praise or criticism, of users and determining whether it is positive or negative or spending the extra money to hire people to do that. Later, it will help identify whether the comment is spam or not, which can reduce time determining feedback's polarity, determine the language of the comment, and so on.

2. Literature Review

Sentimental analysis, which will be used in this research, has been studied in detail for the last few years. There are a lot of research papers regarding sentimental analysis, but we will present only the ones that are useful for our research.

In the paper [3], authors Akanksha Sharma and Dr. Ashim performed a Comparative Study of Different Approaches Used For Sentiment Analysis from customer reviews, where they stated that this process helps the owners of the online shop to make the right decision regarding their items. In their research, they have divided the feedback into three categories: positive, negative, and neutral. Where we can notice that in our research the classification of feedback is similar, from -1 to 1. 1 represents positive, 0 represents neutral and -1 represents negative. Their research is very similar to ours. They gathered feedback from e-shops, analyzed the feedback, and finally classified them. The authors mentioned Support Vector Machine (SVM), Naive Bayes, Lexicon Method, etc. At the end of their research, SVM was the best compared to other methods.

Research paper [4], also performed a sentiment analysis on user feedback from online shops. Michael Gamon, the author of this research, uses over 40.000 feedbacks that he collected from two different sources, Global Support Services, and Knowledge Base Surveys. The author divided the feedback on a scale between 1 and 4, where 1 represented dissatisfied and 4 for very satisfied. In his research, he used a linear Support Vector Machine (SVM) for feedback classification with 10-fold cross-validation. As a result of his research, Michael created two clear classifications (classes). The first class determines

whether the feedback falls under 1 or 4 on the scale, the second class determines whether the feedback falls under 1 or 2 and 3 or 4. He used 10-fold cross-validation on both classifications with the first 2000 feedbacks in his dataset. The first class (whether the feedback falls under 1 or 4 on the scale) proved to be more accurate. The precision was 85.47 for the first class, while the second class was 69.23.

Prashali et al. [5], the authors of the research, collected their research data for the classification from Kaggle website. The data was in excel format, containing 186 feedback. The goal of their research was to see how to improve the teaching and learning program. Their dataset was composed of students' feedback on the teaching program. The result of their research was divided, as in our research, between -1 and 1. As we mentioned before 1 represents positive, 0 neutral, and -1 negative. We have to mention that in their research, they used polarity from sentiment analysis to determine whether the feedback is classified as positive, negative, or neutral.

In the paper [6], the authors wrote about how owners of online stores should analyze every feedback they get in the shortest time possible. Since this affects their further business and cooperation with the seller on their online shop. Robots can cause fraud to star ratings on items on online shops, for that reason feedback on online shops must be analyzed using natural language processing (NLP). In this way, we can delete false feedback and quickly analyze feedback received. Swati N. Manke and Nitin Shivale classify their results in two categories, positive and negative.

Author Peter D. Turney in his research paper [7], applied semantic orientation for determining whether the feedback is positive or negative. For his research, Peter used 410 samples of feedback, which he acquired from 4 different domains (banks, automobile, movie, and travel destination). He used an unsupervised learning algorithm to classify feedback as positive or negative. The precision of his algorithm was averaging 74%, the highest precision was on automobile 84%, while the lowest one was on movie 66%. The reason for the difference between the precision of automobiles and movies, which was a pretty huge one was because of some words depending on the context. In the domain of automobiles, some adjectives may have a negative meaning whereas in the movie sphere it can be the exact opposite meaning. For example, the adjective "unpredictable" would have a negative meaning in an automobile but in the movie a positive one. For assessing feedback to be positive or negative, the author Peter followed 3 steps:

- Draw out sentences which contain adjectives and adverbs,
- Predict semantic orientation of each extracted sentence,
- Categorize feedback as positive or negative according to the semantic orientation of the sentence.

In [8], the authors used a model to analyze text from feedback written by the users in their research. Also, the number of stars of the star rating given by the user was taken for determining the results. Joachim Büschken and Greg M. Allenby tested their model on a hotel and restaurant dataset, which contained the feedback and the star rating. Their model was built based on Latent Dirichlet Allocation (LDA). In the

restaurant dataset, there are 696 samples (feedback and star rating) from different Italian restaurants. While in the hotel dataset, feedback and star ratings were collected from two hotels, one in New York and the second near the JFK airport. The number of samples collected from the hotel in New York is 3.212, while the second hotel is 1.255, which sums up to 4.467 feedback from the hotels. At the end of their research, the authors believe that bag-of-sentence is better than bag-of-words for user speech analysis.

Saleem Abuleil and Khalid Alsamara in their research paper [9], wrote about analyzing user feedback using Natural Language Processing (NLP). The authors presented feedback in two formats, rating (structured data) and textual (unstructured data). Their research was applied on feedback that has been written in the Arabic language. In the Arabic language, adjectives take the form of describing another person or thing in a sentence. In their research, the authors convert unstructured data (text) into structured data (numerical). They categorized their results into two classes, positive and negative feedback.

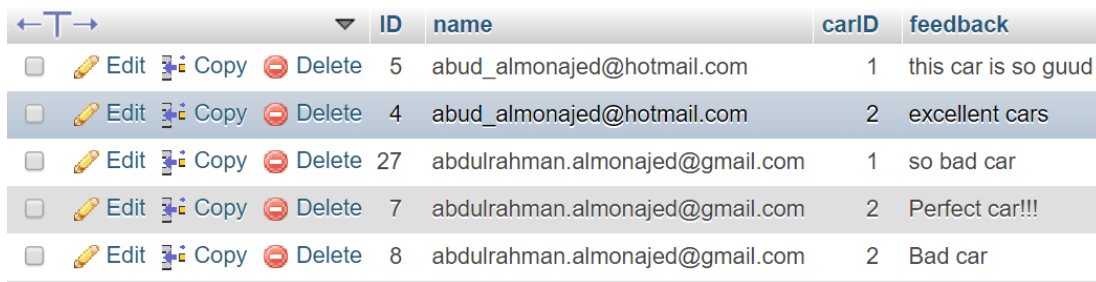
In the research paper [10], authors write about measuring customers' satisfaction using sentiment analysis. For the classification method, they used sentiment classifier support vector machine (SVM). The main reason for that was that SVM gave the best results on the basis of the research paper [11]. The data set was collected from Twitter API. It contained the following:

- Likes (lists of users that liked specific tweet)
- Followers (lists of users that follow specific tweet)
- Mentions (lists of users that was mentioned on a specific tweet)
- Replies (lists of replies on a specific tweet), and
- Re-tweet (lists of users that share specific tweet)

In this research, they used the database MySQL Database Management System. The authors classified their results in two classes, positive and negative. At the end of their research, their algorithm gave a precision of around 87%.

3. Methods and Materials

The data that will be used in this research will be taken from the BTT web application. The number of feedback samples is more than 1000. The application consists of multiple feedback sites, but this research will be based on feedback from the rent-a-car section/site. The number of data we will test in this research will depend on the number of feedbacks on the BTT web application. Right now, there are more than 1000 feedback for the rent-a-car section, if new feedback is added, the system will cover them automatically once it runs. We only used cars' feedback from the BTT web application. We take data in HTML format where we have only feedback, without other attributes from the table that are related to feedback for business logic. The attributes that we will not use are ID, user, and car_ID since it means nothing to us in determining whether the feedback is positive or negative. This means that only one column is left since the table contains 4 columns (ID, name, carID, and feedback), which we can see in the figure below.



| | ID | name | carID | feedback |
|---|----|---------------------------------|-------|---------------------|
| <input type="checkbox"/> Edit Copy Delete | 5 | abud_almonajed@hotmail.com | 1 | this car is so guud |
| <input type="checkbox"/> Edit Copy Delete | 4 | abud_almonajed@hotmail.com | 2 | excellent cars |
| <input type="checkbox"/> Edit Copy Delete | 27 | abdulrahman.almonajed@gmail.com | 1 | so bad car |
| <input type="checkbox"/> Edit Copy Delete | 7 | abdulrahman.almonajed@gmail.com | 2 | Perfect car!!! |
| <input type="checkbox"/> Edit Copy Delete | 8 | abdulrahman.almonajed@gmail.com | 2 | Bad car |

Figure 1. Feedback in MySQL

As we mentioned before, we will only use one column for the table, which is the feedback column. Figure 2 shows only feedback from the table in the HTML web page, from where we will take the feedback.

- this car is so guud
- excellent cars
- so bad car
- Perfect car!!!
- Bad car
- Perfect
- Good
- Very bad

Figure 2. Feedback on HTML page

A. Data preprocessing

Since this research is based on working with text, in the process of determining whether the given text is positive or negative, that text must be analyzed and processed. The system will be based solely on working with English text. We will implement natural language processing (NLP) in the process of further analyzing and processing the feedback. For the whole process, we will use the python programming language with its library TextBlob. The library TextBlob will be used to determine if the given feedback is positive or negative. TextBlob is a python library used for basic text tasks, such as sentiment analysis, translation, language determination, and so on. All of these tasks can be classified under NLP tasks. TextBlob allows us to view objects as a regular string in the python for processing the desired task [12]. The processes and analyzes done in this research are removing HTML tags, removing non-letters, removing whitespaces and empty elements, lowercase, tokenization, spell checking and correct misspelled words, and etc. To reduce the number of words of the feedback and make the classification as accurate as possible, usually removing stopwords is used [13]. When we check the list of nltk's stopwords, we can see that it's not a good idea to always remove stopwords from the dataset. For example, the stopword "not" it can change the meaning of the sentence at all. Since, the sentence "This car is not good" after removing stopwords will be "car good". We can see that the original sentence is

negative, while the sentence after removing stopwords is positive. Of course, it's not always the case that removing stopwords will change the meaning of the sentence. Because of that, before removing stopwords it is good to know the sentences inside the dataset.

The figure below shows the example of how removing stopwords can change the meaning of the sentence.

```
['this', 'car', 'is', 'not', 'good']  
['car', 'good']
```

Figure 3. Example of removing stopwords

In Figure 4. we prove that removing stopwords sometimes can cause an issue. We can see that first sentence as polarity result -35, which means it's negative, while after removing stopwords from the sentence, the meaning is changes and polarity result became 70, which means the sentence is positive.

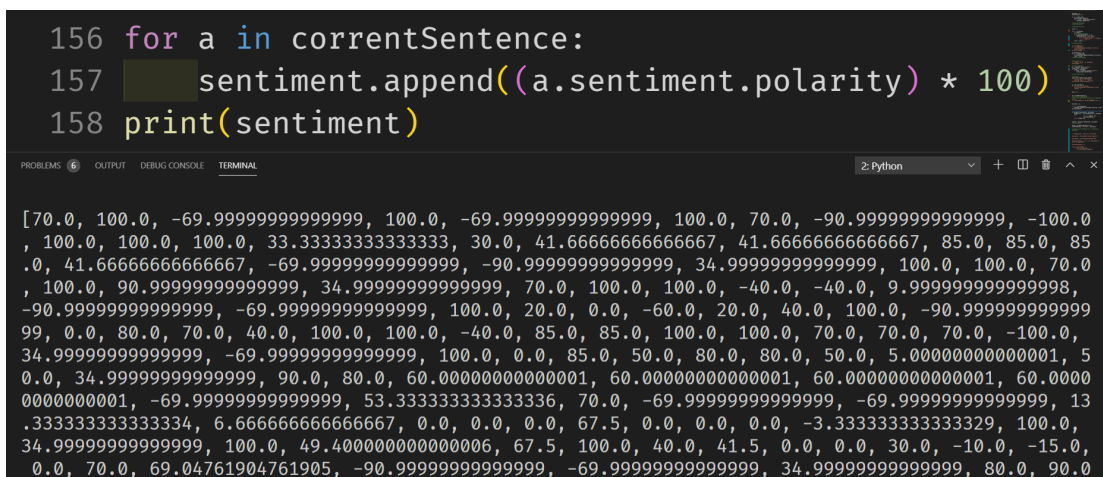
```
[(TextBlob("This car is not good"), -35.0)]  
[(TextBlob("good car"), 70.0)]
```

Figure 4. Polarity result before and after removing stopwords

4. Results

After processing the above analyzes and processes on each feedback we took from the BTT web application, we will begin the process of determining whether it is positive or negative. Here we come to sentiment analysis, which will be used from the mentioned python library. From TextBlob's sentiment analysis, we will use the polarity part which will give us a result between -1 and 1. Where -1 indicates very negative results, in our case very bad feedback, and 1 is positive. In Figure 5, we show the implementation of textblob's sentiment polarity and the polarity result or score.

```
156 for a in correntSentence:  
157     sentiment.append((a.sentiment.polarity) * 100)  
158 print(sentiment)
```



```
[70.0, 100.0, -69.99999999999999, 100.0, -69.99999999999999, 100.0, 70.0, -90.99999999999999, -100.0  
, 100.0, 100.0, 100.0, 33.33333333333333, 30.0, 41.66666666666667, 41.66666666666667, 85.0, 85.0, 85  
.0, 41.66666666666667, -69.99999999999999, -90.99999999999999, 34.99999999999999, 100.0, 100.0, 70.0  
, 100.0, 90.99999999999999, 34.99999999999999, 70.0, 100.0, 100.0, -40.0, -40.0, 9.99999999999998,  
-90.99999999999999, -69.99999999999999, 100.0, 20.0, 0.0, -60.0, 20.0, 40.0, 100.0, -90.99999999999999  
99, 0.0, 80.0, 70.0, 40.0, 100.0, 100.0, -40.0, 85.0, 85.0, 100.0, 100.0, 70.0, 70.0, 70.0, -100.0,  
34.99999999999999, -69.99999999999999, 100.0, 0.0, 85.0, 50.0, 80.0, 80.0, 50.0, 5.000000000000001, 5  
0.0, 34.99999999999999, 90.0, 80.0, 60.000000000000001, 60.000000000000001, 60.000000000000001, 60.0000  
0000000001, -69.99999999999999, 53.33333333333336, 70.0, -69.99999999999999, -69.99999999999999, 13  
.33333333333334, 6.66666666666667, 0.0, 0.0, 0.0, 67.5, 0.0, 0.0, 0.0, -3.33333333333329, 100.0,  
34.99999999999999, 100.0, 49.400000000000006, 67.5, 100.0, 40.0, 41.5, 0.0, 0.0, 30.0, -10.0, -15.0,  
0.0, 70.0, 69.04761904761905, -90.99999999999999, -69.99999999999999, 34.99999999999999, 80.0, 90.0
```

Figure 5. Implementation and result of TextBlob's sentiment property

From the figure above, we can see that the result is not so readable, where we can only check for the polarity result but we don't know for which feedback is that result. So we combined or merged the polarity score and feedback, to make the result more readable and understandable. The figure below, shows the way we combined the feedback and polarity score, and how the result became more readable and understandable from before.

```

160 def merge(correntSentence, sentiment):
161     merged_list = [(correntSentence[i], sentiment[i])
162                   for i in range(0, len(correntSentence))]
163     return merged_list
164
165 combine = merge(correntSentence, sentiment)
166 print(combine)
    
```

```

[(TextBlob("this car is so good"), 70.0), (TextBlob("excellent cars"), 100.0), (TextBlob("so bad car"), -69.99999999999999), (TextBlob("perfect car"), 100.0), (TextBlob("bad car"), -69.99999999999999), (TextBlob("perfect"), 100.0), (TextBlob("good"), 70.0), (TextBlob("very bad"), -90.99999999999999), (TextBlob("terrible cars"), -100.0), (TextBlob("awesome"), 100.0), (TextBlob("perfect car"), 100.0), (TextBlob("excellent car"), 100.0), (TextBlob("super car"), 33.33333333333333), (TextBlob("fun car"), 30.0), (TextBlob("fine car"), 41.66666666666667), (TextBlob("fine"), 41.66666666666667), (TextBlob("beautiful"), 85.0), (TextBlob("beautiful car"), 85.0), (TextBlob("car is beautiful"), 85.0), (TextBlob("car is fine"), 41.66666666666667), (TextBlob("car is bad"), -69.99999999999999), (TextBlob("very bad car"), -90.99999999999999), (TextBlob("this car is not bad"), 34.99999999999999), (TextBlo
    
```

Figure 6. Feedbacks' polarity result

The table below shows the total accuracy of our algorithm.

Table 1. Result

| Algorithm | Approximate result |
|-------------|--------------------|
| Naïve Bayes | ~ 80% |

5. Discussion

Considering the research papers related to our research, which are already mentioned in the Section 2, we have notice that it is much faster and easier to determine if the feedback is positive or negative using the Python's library TextBlob. As we mentioned before, it is not always good idea to remove stopwords from the text, as it can change the meaning of the sentences. In some researches, Naïve Bayes algorithm didn't give the best result. There may be more causes such as, huge dataset with unnecessary sample or information, stopwords are removed, preprocessing is not done properly, and so on. In the table below, we showed the algorithms and results of several previous researches.

Table 2. Conspectus of previous works

| Author(s) | Algorithm | Result |
|---|---|---|
| Joachim Büschken and Greg M. Allenby | LDA (Latent Dirichlet Allocation) | 60-70% |
| Michael Gamon | SVM (Support Vector Machine) – two classes | 85.47% for the first class, while the second class was 69.23% |
| Al-Otaibi Shaha, Alnassar Allulo, Alshahrani Asma, Al-Mubarak Amany, Albugami Sara, Almutiri Nada, Albugami Aisha | SVM (Support Vector Machine) | Around 87% |
| Peter D. Turney | PMI-IR (Pointwise Mutual Information - Information Retrieval) | Around 74% |

6. Conclusion

To conclude the results, the feedback has been divided into two groups, positive and negative. Feedback, like in every web site helps the users that are first time on the online shop to determine which product is of good quality. In this research we proved that removing stopwords is not always a good idea, because it can change the meaning of the sentence. Also the research will make it easier for the online shop owners to determine which feedback is positive and which is negative. In this way the owner will be able to recognize the quality sellers in a very easy and simple way. In the near future we are planning to improve this research by adding 'minus'. The minus will be added to sellers for every bad/negative feedback on his items. In that way we will be able to isolate bad sellers with bad items. If the seller receives a certain number of minuses he will be warned. If the seller's item gets a certain amount of minuses it will be automatically deleted. Also a method for recognising whether a feedback is spam or not will be implemented. This process will be initiated before the sentimental analysis. Since we want to perform the sentimental analysis only on „real“ feedback. This will speed up the process because we will not analyse large numbers of spam feedback. Also methods for translating foreign feedback to English language will be added. This research will be open-source so that every company or person will be able to use it, of course they will need to own a shop which receives feedback.

REFERENCES

| | |
|------|---|
| [1] | S. CK i G. Edwin, "Online Shopping - An Overview," June 2014. [Na mreži]. Available: https://www.researchgate.net/publication/264556861_Online_Shopping_-_An_Overview . |
| [2] | A. Fundin i B. Bergman, "Exploring the customer feedback process," June 2003. [Na mreži]. Available: https://www.researchgate.net/publication/240260148_Exploring_the_customer_feedback_process . |
| [3] | A. Sharma i A. Dr. Saha, "A comparative Study of different Approaches Used for Sentiment Analysis From Customer Reviews," 14 Dec 2018. [Na mreži]. Available: https://poseidon01.ssrn.com/delivery.php?ID=5751021240710700920281070710870230680370490040060050301200900180960751141181190270711060980510290180320020670021111090060881061220260940480650751111250880150870891260690020340740060171160050860910251130010930131 . |
| [4] | M. Gamon, "Sentiment classification on customer feedback: Noisy data, large feature vectors, and the role of linguistic analysis," January 2004. [Na mreži]. Available: https://www.researchgate.net/publication/215470705_Sentiment_classification_on_customer_feedback_data_Noisy_data_large_feature_vectors_and_the_role_of_linguistic_analysis . |
| [5] | S. S. Prashali , R. K. Asmita , S. P. Rutuja i U. W. Yamini , "Sentiment Analysis of Feddback Data," March 2019. [Na mreži]. Available: https://www.ijtsrd.com/papers/ijtsrd23090.pdf . |
| [6] | N. M. Swati i Nitin Shivale, "A Review onL Opinion Mining and Sentiment Analysis based on Natural Language Processing," <i>International Journal of Coumputer Applications</i> , pp. 29-32, 2015. |
| [7] | D. T. Peter, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," July 2002. [Na mreži]. Available: https://www.aclweb.org/anthology/P02-1053.pdf . |
| [8] | J. Büschken i G. M. Allenby, "Sentence-Based Text Analysis for Customer Reviews," 2016. [Na mreži]. Available: https://www.ku.de/fileadmin/160102/WiSe2015_2016/mksc.2016.0993-ePDF3.pdf . |
| [9] | S. Abuleil i K. Alsamara, "Using NLP Approach for Analyzing Customer Reviews," 2018. [Na mreži]. Available: https://www.slideshare.net/cscpcconf/using-nlp-approach-for-analyzing-customer-reviews-86265367 . |
| [10] | S. Al-Otaibi, A. Alnassar, A. Alshahrani, A. Al-Mubarak, S. Albugami , N. Almutiri i A. Albugami, "Customer Satisfaction Measurement using Sentiment Analysis," <i>International Journal of Advanced Computer Science and Application</i> , pp. 106-117, 2018. |
| [11] | J. Brynielsson, F. Johansson, C. Jonsson i A. Westling, "Emotion classification of social media posts for estimating people's reactions to communicated alert messages during crises," 2014. [Na mreži]. Available: https://docplayer.net/11592731-Emotion-classification-of-social-media-posts-for-estimating-people-s-reactions-to-communicated-alert-messages-during-crises.html . |
| [12] | S. Loria, "textblob Documentation," 26 April 2020. [Na mreži]. Available: https://buildmedia.readthedocs.org/media/pdf/textblob/latest/textblob.pdf . |
| [13] | S. Bird, E. Klein i E. Loper, <i>Natural Language Processing with Python</i> , O'REILLY, 2009. |