# PYTHIA: Employing Lexical and Semantic Features for Sentiment Analysis

Ioannis Manoussos Katakis[1], Iraklis Varlamis[1], and George Tsatsaronis[2]

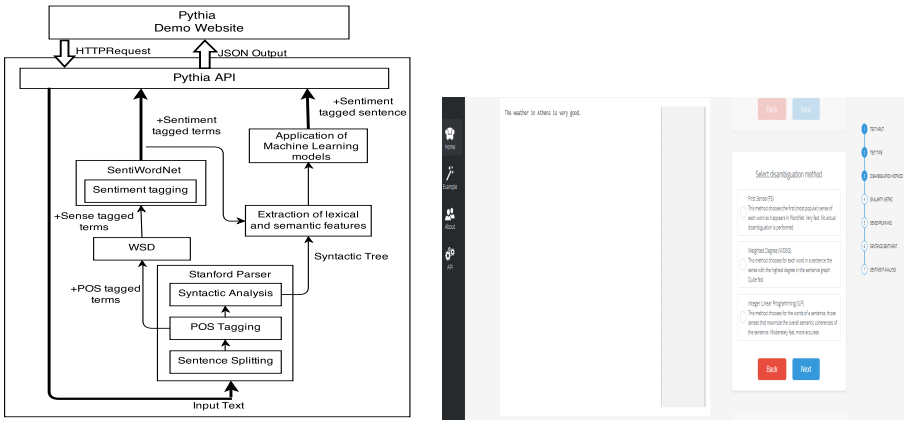
[1] Department of Informatics and Telematics, Harokopio University, Athens, Greece
imktks@gmail.com, varlamis@hua.gr
[2] Biotechnology Center, Technische Universität Dresden, Dresden, Germany
george.tsatsaronis@biotec.tu-dresden.de

**Abstract.** Sentiment analysis methods aim at identifying the polarity of a piece of text, e.g., passage, review, snippet, by analyzing lexical features at the level of the terms or the sentences. However, many of the previous works do not utilize features that can offer a deeper understanding of the text, e.g., negation phrases. In this work we demonstrate a novel piece of software, namely PYTHIA[3], which combines semantic and lexical features at the term and sentence level and integrates them into machine learning models in order to predict the polarity of the input text. Experimental evaluation of PYTHIA in a benchmark movie reviews dataset shows that the suggested combination performs favorably against previous related methods. An online demo is publicly available at http://omiotis.hua.gr/pythia.

## 1 Introduction

Addressing sentiment analysis as a machine learning problem, e.g., as text classification, poses certain challenges, such as the large space complexity, and the need for deeper understanding of the text, which has given rise to deep learning techniques [1]. In this paper we address the task of sentiment analysis as a binary classification problem; positive or negative polarity. The novelty of our approach lies in the integration of lexical features, e.g., term n-grams, which have been used in the past by previous approaches [1] with semantic features, e.g., count of terms with positive and negative polarity at the sentence level. For the extraction of the semantic features we employ novel word sense disambiguation techniques. Finally, we analyze systematically the effect of all feature types, and we evaluate experimentally our approach by using a variety of machine learners for the task. Comparative evaluation with previous works in a movie review benchmark dataset shows that the suggested approach compares favorably against the previously reported results. The resulting approach is offered as an online system which is publicly available (http://omiotis.hua.gr/pythia) and customizable. The users may select among different disambiguation methods, machine learners and feature types, and can test the approach with any

[3] PYTHIA (pronounced pɪθɪə), was the priestess at the Oracle of Delphi. The story says that PYTHIA spoke gibberish, which was then interpreted by the priests.




(a) Pipeline of PYTHIA.

(b) Example of the WSD options.

**Fig. 1.** Overview of the pipeline adopted by PYTHIA and an example of the user options offered.

piece of text as input. The output of the system is the prediction of the overall polarity of the text, and the annotation and highlighting of the text fragments with information that played important role in the final polarity decision. As a result, the presented online system is of great value to researchers of the field, as well as practitioners who aim at utilizing sentiment analysis approaches in wider text processing software components.

## 2 Methods, Evaluation and Demonstration of PYTHIA

The overview of the processing pipeline of the PYTHIA system is shown in Fig. 1(a). PYTHIA implements five main components: (1) syntactic analysis of input text, (2) word sense disambiguation (WSD), (3) tagging of terms with sentiment labels, (4) extraction of lexical and semantic features, and, (5) application of machine learning models to predict the polarity of the input text. For several of these components the API of the online system offers multiple alternatives that can be customized by the user, e.g., as Fig. 1(b) shows for the WSD options.

For the syntactic analysis PYTHIA employs the *Stanford Parser*[4]. The output of this step is a set of trees (one for each sentence) annotated with part of speech (POS) information for each term. The POS information is useful for the WSD step that follows. For the WSD component, PYTHIA implements three options: (1) the first sense heuristic, that always selects the most frequent sense for each word based on *WordNet*, (2) a graph based disambiguation technique, called weighted degree (WDEG) [2], which is a version of Degree Centrality for weighted graphs, and, (3) an Integer Linear Programming approach ILP [3] which solves the ILP problem of maximizing the total pairwise relatedness of

[4] `http://nlp.stanford.edu/software/lex-parser.shtml`

| ML | Semantic Features (40) | Char $n-$grams (11,923) | Term $n-$grams (214,342) | All $n-$grams (225,475) | All Features (225,515) |
|---|---|---|---|---|---|
| SVM | 68.26 | 73.35 | **80.11** | 79.01 | 80.04 |
| Logistic Regression | **68.43** | 69.07 | 77.31 | 78.65 | 79.01 |
| Naive Bayes | 64.66 | **75.35** | 74.32 | **79.81** | **80.73** |

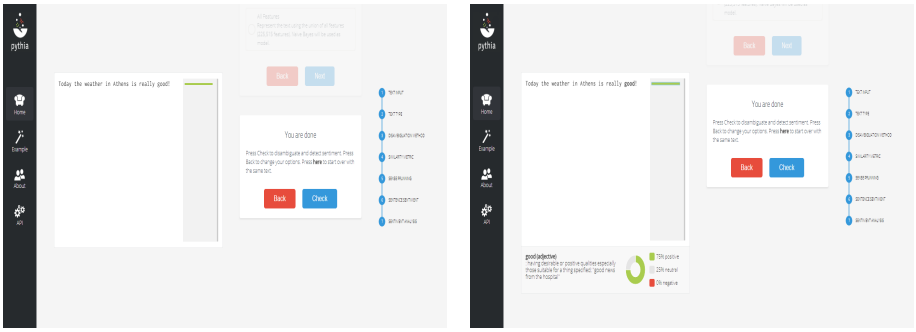
**Table 1.** Overall accuracy obtained at the test set of the movie reviews dataset in predicting positive or negative polarity, per feature type and machine learner used. In parenthesis the number of features is reported.

the selected senses. For the latter WSD approach, any sense relatedness measure can be used; PYTHIA is using three alternative measures: the knowledge-based *SR* measure [4], the corpus-based point-wise mutual information (*PMI*) and a Lesk-like hybrid measure [5]. For the tagging of terms with sentiment labels PYTHIA finds the polarity of each disambiguated word using *SentiWordNet*[5]. *SentiWordNet* is a lexical resource, which assigns to each synset of *WordNet* sentiment scores for positivity, negativity and objectivity. Next, PYTHIA uses the output of the previous components to extract semantic and lexical features for the input text. The semantic features employed by PYTHIA are 40 and are of two types: (i) at the term level, they capture the number, type and polarity score of the terms that contribute some sentiment to the sentence, and, (ii) at the phrases level they capture the same information as before, but by analyzing whole phrases of the sentence instead of terms. Examples of the semantic features are: the number of nouns with positive polarity in the sentence, the total positive polarity score of verbs, and, the number of noun phrases with negative polarity. In addition to the semantic features, PYTHIA also employs two types of lexical features, namely character and term $n-$grams, with $n = [1, 3]$. The final step is the application of a machine learning model using some or all of the aforementioned features, in order to predict the polarity of the input text (positive or negative).

The selection of the offered classifiers in PYTHIA is based on the results of the comparative experimental evaluation we conducted. For this purpose, we used a benchmark dataset in sentiment analysis that contains $9,613$ sentences from movie reviews. We used the split into training $(7,792)$ and test $(1,821)$ introduced in [1]. Table 1 shows the results of the evaluation, reporting only on the top-3 tested classifiers (Support Vector Machines, Naive Bayes, and Logistic Regression). The top accuracy obtained for each of the feature types is highlighted, reaching up to 80.73% when all of the features are used, and Naive Bayes is used as a learner. These results are comparable with the SoA results presented in [1], where the authors report 79.4% for the SVM, and 81.8% for the Naive Bayes using BoW representations of the text. The key finding of the experimental evaluation, which constitutes the novelty of the PYTHIA approach, is that the combination of all features, semantic and lexical, leads to the best results.

Finally, in Fig. 2 we present screenshots of the PYTHIA demo. Fig. 2(a) shows the results of the sentiment analysis of an input sentence, which are pre-

[5] `http://sentiwordnet.isti.cnr.it/`



(a) Results in PYTHIA. (b) Mouse-over functionality.

**Fig. 2.** Screenshots of the PYTHIA demo.

sented with a user-friendly GUI. The used model is automatically selected based on the selection of the feature types, e.g., if all features are selected the Naive Bayes classifier is used. Fig.2(b) shows the response of PYTHIA when the user places her mouse over the sentence terms; the polarity score of the specific term is shown. In addition to this, all PYTHIA features are publicly available via an API that uses GET and POST methods which return *JSON* objects[6].

## 3 Summary

In this article we presented PYTHIA, a demo for sentiment analysis which employs semantic and lexical features in order to predict the sentiment of an input text. Evaluation of PYTHIA in a benchmark dataset with movie reviews showed that the implemented methods may achieve an accuracy of up to 81%, and that the combination of the semantic and lexical features provided the best performing set up.

## References


1. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP, ACL (2013) 1631–1642
2. Sinha, R., Mihalcea, R.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: Proc. of the IEEE ICSC. (2007) 363–369
3. Panagiotopoulou, V., Varlamis, I., Androutsopoulos, I., Tsatsaronis, G.: Word sense disambiguation as an integer linear programming problem. In: SETN 2012. LNCS (7297), Springer (2012) 33–40
4. Tsatsaronis, G., Varlamis, I., Vazirgiannis, M.: Text relatedness based on a word thesaurus. JAIR **37** (2010) 1–39
5. Nguyen, K., Ock, C.: Word sense disambiguation as a travelling salesman problem. AI Review (2011) 1–23


[6] `http://omiotis.hua.gr/pythia/api.html`