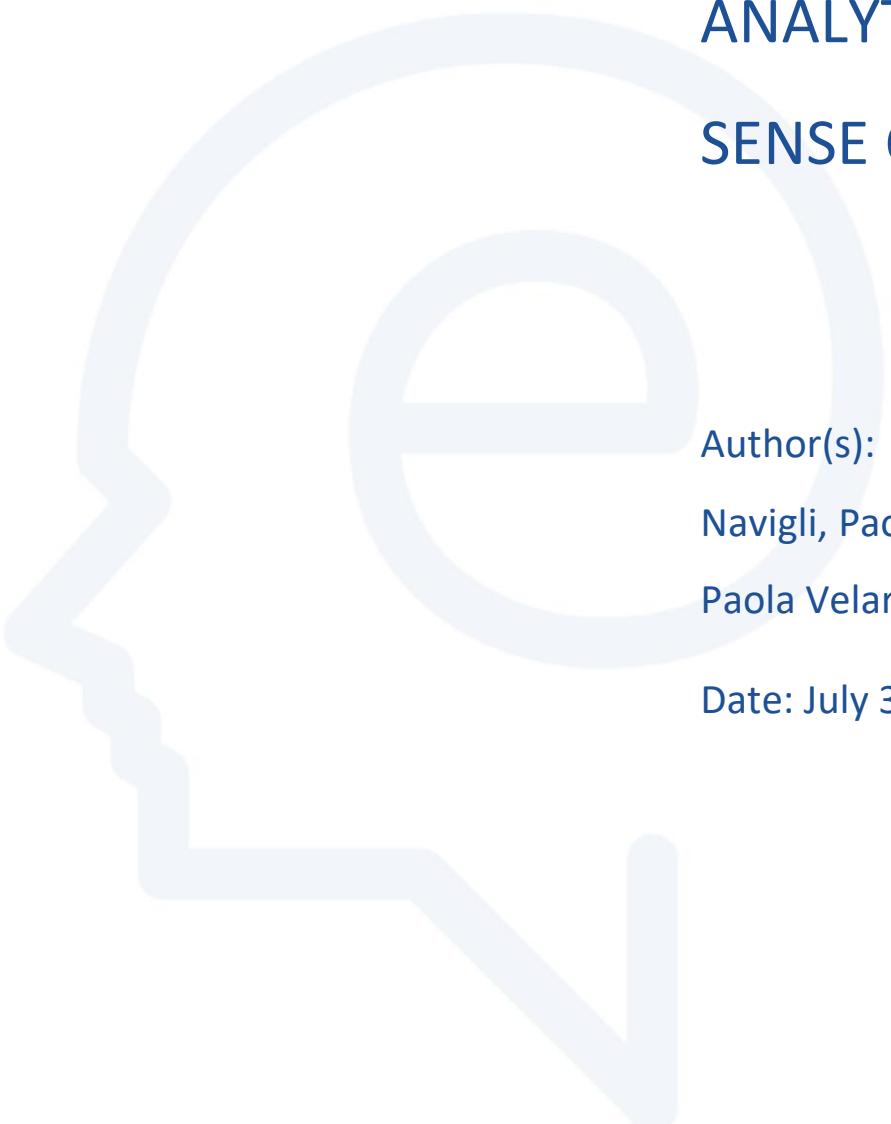


D3.1

LEXICAL-SEMANTIC ANALYTICS FOR NLP: SENSE CLUSTERING



Author(s): Federico Martelli, Roberto Navigli, Paolo Spadoni, Giovanni Stilo, Paola Velardi

Date: July 30, 2019

H2020-INFRAIA-2016-2017

Grant Agreement No. 731015

ELEXIS - European Lexicographic Infrastructure

D3.1 LEXICAL-SEMANTIC ANALYTICS FOR NLP:
SENSE CLUSTERING

Deliverable Number: D3.1

Dissemination Level: Public

Delivery Date: 31/07/2019

Version: Final

Author(s): Federico Martelli,
Roberto Navigli, Paolo
Spadoni, Giovanni Stilo,
Paola Velardi.

Project Acronym: ELEXIS
Project Full Title: European Lexicographic Infrastructure
Grant Agreement No.: 731015

Deliverable/Document Information

Project Acronym: ELEXIS
Project Full Title: European Lexicographic Infrastructure
Grant Agreement No.: 731015

Document History

Version Date:	Changes/Approval:	Author(s)/Approved by:
25 July 2019	Initial version	Federico Martelli, Roberto Navigli
27 July 2019	Assessment by	Simon Krek
30 July 2019	Final version	Federico Martelli, Roberto Navigli

D3.1 Lexical-semantic analytics for NLP: Sense clustering.

Table of Contents

1	INTRODUCTION	3
2	SENSE CLUSTERING	4
2.1	Task overview.....	4
2.2	Task description	4
3	APPROACH	5
3.1	Lexical vector representation	5
3.2	Sense clustering algorithm.....	5
4	EVALUATION SETUP	7
4.1	In vitro evaluation	7
4.2	In vivo evaluation	7
5	RESULTS.....	8
5.1	In vitro results	8
5.2	In vivo results	8
6	INSTRUCTIONS	10
7	REFERENCES	13



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

List of Tables

Table 1.....	8
Table 2.....	9
Table 3.....	9



1 INTRODUCTION

The goal of task 3.3, *Lexical Semantic Analytics for NLP*, which the present deliverable is part of, is to discover and investigate new approaches for performing **text analytics** for words, senses, domains and other lexical-semantic information such as phrases and collocations. To this aim, the task proposes to explore three directions: **sense clustering**, **domain-labeling of texts** and **diachronic distribution of senses**.

This deliverable focuses on the first subtask and presents an innovative algorithm aimed at addressing one of the most challenging NLP tasks, namely **sense clustering**. Despite recent advances in neural architectures, we decided to further investigate alternative approaches which allow to effectively and easily scale across languages and, most importantly, drop the requirement of large amounts of data which is typically needed when employing neural networks and which would hamper the applicability of sense clustering to arbitrary dictionaries and languages.

The present deliverable is structured as follows. First, we will introduce the reader to the task of sense clustering. Second, we will illustrate the algorithm employed. Third, we will describe our in vitro and in vivo evaluation, demonstrating the efficacy of our algorithm when performing a crucial NLP task, namely Word Sense Disambiguation (WSD).



2 SENSE CLUSTERING

2.1 Task overview

It has been widely demonstrated that too fine-grained sense distinctions negatively affect the performance of several computational tasks such as WSD whose objective is to automatically identify the correct meaning of a word in a given context. Typically, such fine granularity can be observed in the vast majority of lexical-semantic resources employed for computational purposes such as WordNet, a wide-coverage lexicon of the English language. Over the last years, several efforts have been made for the purposes of reducing the number of subtle sense distinctions in sense inventories. Such task takes the name of sense clustering and has gradually become one of the most challenging Natural Language Processing (NLP) tasks.

2.2 Task description

Sense clustering is the computational task of partitioning a set of senses into subsets consisting of semantically-related senses, called clusters. The number of elements included in each cluster is variable and depends on the degree of granularity to be obtained. The major goal of sense clustering is to reduce the fine granularity of a sense inventory which plays a fundamental role in a wide range of NLP tasks. For instance, in the field of WSD, it has been shown that a coarse-grained sense inventory can lead to a remarkable increase in performance (Navigli, 2009). Particularly surprising is the fact that even human annotators seem to be often in disagreement when asked to identify, within a fine-grained inventory, the most appropriate sense of an ambiguous word. In these cases, as shown in the literature, the inter-annotator agreement ranges between 0.6 and 0.8. This is a clear indication that fine granularity does not rest on solid foundations and that sense clustering is required in order to achieve an optimum balance between performance and ambiguity.



3 APPROACH

In this Section, we will detail the approach that we adopted for computing sense clustering. We first describe how to create a vector representation for each dictionary definition and then illustrate the clustering algorithm.

3.1 Lexical vector representation

Starting from a dictionary definition of a word meaning, our algorithm consists of the following steps:

1. We **extract the content words** from the definition and create a bag of words; if the sense entry provides additional information (e.g. its hypernym, usage information), the corresponding content words are also added;
2. For each bag of words, we query an information retrieval system with the extracted content words and **retrieve the Wikipedia pages which are most relevant to the target word sense**. We use Lucene to retrieve the top-scoring Wikipedia pages.
3. We **compute a lexical vector of the target sense by concatenating the words contained in the Wikipedia pages retrieved**. Each component of the vector represents a word in the vocabulary and is valued with its relevance for the target sense. Such relevance is computed in terms of its lexical specificity (Lafon, 1980) against the whole Wikipedia corpus.

3.2 Sense clustering algorithm

After obtaining vector representations for each sense of an ambiguous word, we perform sense clustering as follows:

1. We **calculate the similarity between all vector representations** encoding all sense pairs;
2. We **order these pairs by their cosine similarity score**;



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

- We determine empirically three different thresholds** which are aimed at identifying which senses have to be included in the same cluster.

In the following we show the pseudocode which implements the above three steps:

```
p = compute all possible sense pairs of a given word w
sort these pairs by score in descending order
```

```
clusters = []
```

```
For each pair p ranked by score s:
```

```
  if s > t1:
    c1 = find the cluster of sense s1 of p
    c2 = find the cluster of sense s2 of p

    if c1 is null and c2 is null
      c3 = a new cluster
      add s1 to c3
      add s2 to c3
      add c3 to clusters

    else if c1 is null and c2 is not null:
      sim = compute the average similarity between s2 and all
senses of c1
      if sim > t2:
        add s2 to cluster c2
      else:
        c3 = create a new cluster
        add s2 to c3
        add c3 to clusters
    else if...
      // the same as previous point with swapped variables

    else if c1 is different from c2:
      sim1 = compute the average similarity between s1 and all
senses of c2
      sim2 = compute the average similarity between s2 and all
senses of c1
      if sim1 > t3 and sim2 > t3:
        merge c2 into c1
        remove c2 from clusters
```

```
For all remaining senses create a singleton cluster and add them to
clusters
```



4 EVALUATION SETUP

4.1 In vitro evaluation

In order to evaluate the correctness of our automatic clustering, **we compared the output of our algorithm with a gold standard obtained by manually clustering the WordNet senses of several hundred words.** To this end, we randomly selected a sample of 300 ambiguous words with a polysemy degree ranging between 3 and 10. The manual clustering was performed by expert linguists who brought together senses which are related via systematic polysemy and pertain to the same semantic field.

For each word, we considered all its possible sense pairs and verified whether the senses are included in the same cluster both in our clustering and in the gold standard. We computed the macro accuracy and the micro accuracy. Additionally, as customary when evaluating a sense clustering system, we calculated the improvement of our sense clustering against a random clustering of the same size for each word.

4.2 In vivo evaluation

In this Section we present the evaluation setup which we used for assessing the performance of our algorithm within an application. To perform our in vivo evaluation, we compared the performance of a state-of-the-art neural WSD system (Vial et al. 2019) when employing the standard fine-grained inventory from WordNet and a number of coarse-grained sense inventories, including the one produced with our algorithm. We computed two measures, namely the F1 score and the perplexity which reflects the overall performance of the algorithm and the difficulty at predicting the output, respectively. Furthermore, in order to compute a trade-off figure that summarizes measures with different ranges and magnitudes, we computed a geometric mean of F1 and perplexity.



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

5 RESULTS

5.1 In vitro results

In Table 1, we show the results of the comparison of our automatic clustering against the one obtained manually. The test aims at determining to which degree our clusters are semantically correct.

Table 1. In vitro evaluation against our gold standard of 300 words

Metric	Scores
Macro accuracy	0.714
Micro accuracy	0.730

5.2 In vivo results

In Table 2, we show the results of our in vivo evaluation. We compared the performance of a neural WSD system (Vial et al. 2019) when using the following sense inventories:

1. The WordNet fine-grained sense inventory;
2. Lexnames which is a clustering based on WordNet's lexicographer's ID (<https://wordnet.princeton.edu/>);
3. WordNet domains (Magnini and Cavaglià, 2000), a set of 200 labels loosely following the Dewey Decimal Classification system, and
4. Our coarse-grained sense inventory.

As can be seen in the Table, our sense inventory achieves almost 90% F1 in the WSD task, equalling the results of WordNet domains which is a manual labelling of WordNet synsets. In contrast, our approach is completely automatic.



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

Table 2. In vivo evaluation.

Inventory	F1	PPL	geometric mean
fine-grained	80.5	2.45	14.05
lexnames	87.5	1.86	12.77
WordNet domains	89.9	1.89	13.05
our coarse-grained sense inventory	88.9	1.89	12.97

As customary when evaluating a sense clustering system, we show the improvement against a random clustering, as reported in Table 3.

Table 3. Improvement against a random clustering

Inventory	F1	PPL	geometric mean
our coarse-grained sense inventory	88.9	1.89	12.97
random sense inventory using the same number of clusters for each word and the same number of senses inside each cluster	86.4	1.88	12.76



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

6 INSTRUCTIONS

The present algorithm takes an input lemma *l* and its part of speech (pos, to be chosen among nouns (n), adjectives (a), adverbs (r) and verbs (v)) or all parts of speech (all) and returns a clustering of senses of *l*. Alternatively, the system can be run on all words in WordNet by specifying --all and the pos of interest. In order to run the algorithm, please open a terminal and enter the following instruction:

```
java -jar clusty-1.0.jar <lemma>|--all [n,v,r,a,all]
```



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

7 ELEXIS GitHub repository

The code is available at: <https://github.com/elexis-eu/D3.1>

 README - Clusty v1.0

Linguistic Computing Laboratory
 Sapienza University of Rome
<http://lcl.uniroma1.it>

This package consists of a piece of software for sense clustering based on the WordNet sense inventory. The approach leverages the knowledge contained in Wikipedia for building NASARI lexical vectors.

 CONTENTS

This package contains the following main components:

```

clusty-1.0.jar           # Jar of Clusty
clusty-1.0_lib/         # third party libraries
LICENSE                 # Clusty's license
README                 # this file
pom.xml                 # Maven pom file
run-clustydemo.sh      # shell script to test Babelfy in Linux
  
```

 REQUIREMENTS

We assume that you have a standard installation of the Oracle Java 1.8 JDK and all the associated programs (i.e., java, javac, etc.) in your path.

 INSTALLATION

In order to use Clusty, it is necessary to download WordNet 3.0 and include it in wordnet-releases folder. Furthermore, it is necessary to download, unzip the NASARI lexical vectors here:
https://drive.google.com/file/d/1HqdnFZu__6aAids9p5di8vhURNrhw8Xw/view?usp=sharing
 and include them into the resources folder which needs to be created at the root level.

For testing purposes we provide a shell script:

```

Linux:  run-clustydemo.sh, make sure that the file is
        executable by running: chmod +x run-clustydemo.sh
  
```



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

In order to run Clusty, please specify the following instruction:
`java -jar clusty-1.0.jar <lemma>|--all <n,a,r,v,all>`

AUTHORS

Federico Martelli, Sapienza University of Rome
(martelli@di.uniroma1.it)

Roberto Navigli, Sapienza University of Rome
(navigli@di.uniroma1.it)

Acknowledgments go to Dario Montagnini, Babelscape
(montagnini@babelscape.com), for his contribution to the project.

COPYRIGHT

Clusty is licensed under a Creative Commons Attribution-Noncommercial-
Share Alike 4.0 License. See the LICENSE file for details.

ACKNOWLEDGMENTS

Clusty is an output of the ELEXIS project (<https://elex.is>). This project
has received funding from the European Union's Horizon 2020 research and
innovation programme under grant agreement No. 731015.



D3.1 Lexical-semantic analytics for NLP: Sense clustering.

8 REFERENCES

Lafon, Pierre. 1980. *Sur la variabilite de la frequence des formes dans un corpus*. *Mots*, 1:127–165.

Magnini, Bernardo and Gabriela Cavaglia. 2000. *Integrating Subject Field Codes into WordNet*. In Proc. of LREC, pages 1413–1418.

Navigli, Roberto. 2009. *Word Sense Disambiguation: A Survey*, ACM computing surveys (CSUR), 41(2), 10.

Vial, Loïc, Benjamin Lecouteux, and Didier Schwab. 2019. *Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation*. *arXiv preprint arXiv:1905.05677*.

