# Disclosing Citation Meanings for Augmented Research Retrieval and Exploration

Roger Ferrod[1], Claudio Schifanella[1](✉), Luigi Di Caro[1], and Mario Cataldi[2]

[1] Department of Computer Science, University of Turin
roger.ferrod@edu.unito.it,
{schi,dicaro}@di.unito.it
[2] Department of Computer Science, University of Paris 8
m.cataldi@iut.univ-paris8.fr

**Abstract.** In recent years, new digital technologies are being used to support the navigation and the analysis of scientific publications, justified by the increasing number of articles published every year. For this reason, experts make use of on-line systems to browse thousands of articles in search of relevant information. In this paper, we present a new method that automatically assigns meanings to references on the basis of the citation text through a Natural Language Processing pipeline and a slightly-supervised clustering process. The resulting network of semantically-linked articles allows an informed exploration of the research panorama through semantic paths. The proposed approach has been validated using the ACL Anthology Dataset containing several thousands of papers related to the Computational Linguistics field. A manual evaluation on the extracted citation meanings carried to very high levels of accuracy. Finally, a freely-available web-based application has been developed and published on-line.

**Keywords:** Citation Semantics · Literature Exploration · Natural Language Processing

## 1 Introduction

With more than a million scientific articles published each year, keeping abreast of research progress is becoming an increasingly difficult task. For this reason, an increasing number of scientists rely on digital technologies that can analyze thousands of publications in a short time and provide research support.

Over the last few years, numerous techniques have been developed to analyze large amounts of data: from petabytes produced by the Large Hadron Collider, to the hundreds of millions of bases contained in the human genome; but this vast collection of data has the advantage, unlike natural language, of being naturally represented by numbers which, it is well known, can be easily manipulated by computers. Research literature, on the other hand, seems to be immune to this type of analysis, as the articles are designed to be read by humans only.

Text Mining and Natural Language Processing techniques aim to break this barrier. Using the recent scientific developments of the last thirty years, programs are now learning to extract information from textual sources. This opens up the possibility for scientists to make new discoveries by analyzing hundreds of scientific articles looking for correlations. Among these objectives, there is certainly the discovery of hidden associations, such as links between topics or between authors. Usually, it is possible to search for documents through keywords, or with the use of complex information on a structured database. However, suggestions about articles related to specific studies are extremely helpful in the analysis of the literature.

Of particular interest is the analysis of the citational aspects, i.e., how an article is cited and for which purpose. For this reason, it might be useful to classify citations in classes so as to be able to provide research paths suggesting articles according to specific characteristics or aims.

This work aims at illustrating a novel method for the construction of a semantically-enriched citation graph that makes use of Natural Language Processing and Data Mining technologies to enable advanced retrieval and exploration of a scientific literature. We first tested the approach with a large collection of articles related to the Computational Linguistics field, making available a web-based application called *CitExp* at `http://citexp.di.unito.it`. Finally, we also made a manual evaluation of 900 randomly-selected citation meanings, obtaining very high accuracy scores.

## 2 Related Work

The idea of exploring text collections received huge attention since large datasets started to become available for research purposes. While there exist several approaches for generic navigation objectives, like [2, 20, 18] to name a few, the peculiarity of scientific articles enables further possibilities of semantic access, relying on domain-centered metadata such as citations, co-authoring information, year of publication, and a more rigorous and section-oriented formatting of the content.

In relation with our proposed method, we find that the existing works lie around three main perspectives: *i)* metadata-based navigation and interactive visualization models [2, 18], *ii)* topic- or author-centered browsing and structuring techniques [24, 23], and *iii)* network-based approaches dealing with citations [27, 11, 12], co-authorships [19, 17], and statistically-relevant links between research articles [15, 22]. While we locate our work on the third case, it inherits features from the others while differing from all three on the semi-supervised extraction of semantically-enriched citation paths for an informed navigation of scientific articles.

In recent years, different works offering a navigational model to visually explore scholarly data have been proposed. Faceted-DBLP [5] introduced a facet-based search interface for the DBLP repository, while in PaperCUBE [3] the authors used citations and co-authorship networks to provide browsing functionalities of scientific publications. Similar approaches are proposed by CiteSeer [13], Elsevier Scival and Microsoft Academic Search: in all these proposals, visual interfaces are built on top of a navigational model created from the collaboration networks and the citations. Finally, in the Delve system [1] common browsing functionalities are implemented over a dynamic set of publications, moving the focus on dataset retrieval.

A work similar to ours, based however on the use of ontologies, has been proposed in [25], where the authors identified and formalized different types of citations in scientific articles. In spite of this, the ontology includes a wide set of complex cases, making it exclusively suitable for manual (and costly) annotations of individual references. In [9], it has been developed a graph of publications, grouped according to citation reports and accessible through tools such as CitNetExplorer [8] and VOSviewer [7], where users can perform bibliometric research. In [6], the authors proposed a comparison between various methodologies developed for the purpose of grouping citations into the network. [26] presented a specific semantic similarity measure for short texts such as abstracts of scientific publications. The measure of semantic similarity can be useful to deepen the knowledge on the network of articles to identify groups of publications dealing with similar issues.

## 3 Data and Preprocessing Tools

### 3.1 Data

To test the method we used the ACL Anthology Dataset[3], i.e., a corpus of scientific publications sponsored by the Association for Computational Linguistics. The corpus consists of a collection of articles from various conferences and workshops, as well as past editions of the Computational Linguistics Journal or events sponsored by ACL. The purpose of this anthology does not end with the simple collection of results and publications, but rather arises as an object of study and research platform. Thousands of articles have been digitized and assembled in PDF format, while titles and authors have been extracted from the text to compose the corpus metadata. The current version has 21,520 articles and 287,130 references, of which 91,931 (32%) refer to articles inside the corpus. The ACL anthology is composed of a set of XML files, divided into folders and corresponding to the output of the Parscit software [10].

### 3.2 Snippet Extraction

The basic unit of our work is the *snippet*. A snippet is a portion of text that contains the reference, enriched by other information such as, for example, the section of the article in which it appears. The first phase has the main objective of extracting the snippets from the corpus. Unfortunately, the XML files used as input are devoid of documentation and of a scheme that defines them. Furthermore, some XML elements raise exceptions during the parsing, and the absence of documentation has not facilitated the resolution of errors. For this reason, the

[3] `https://acl-arc.comp.nus.edu.sg`

files causing problems have been deliberately discarded. However, they represent a minimal part of the total corpus (1.73%) and therefore do not compromise the validity of the results. The final result of the preprocessing phase is a single XML file. It is important to note that during the processing by Parscit, the text undergoes various modifications. On the basis of the information obtained from the Parscit source code, it was possible to identify the applied transformations. In particular the excess of white spaces and the hyphenation of the words were removed. If this standardization process were not taken into account, the references (position indexes) indicated by Parscit would not correspond to the actual text.

### 3.3 Snippet Linking

The citations refer, through a numeric id, to *reference* tags containing the details of the referenced article. Parscit is able to extract information such as title, authors, year of publication and other information where present, from the references section of the article. In order to build a graph of articles, however, it is necessary to find a correspondence between the title specified in the reference and the title of the article, since in 23% of the cases these two strings do not coincide. Consider the following example: the title reported by the reference "*an efficient adaptable system for interpreting natural language queries*" actually refers to the article whose title is "*an efficient and easily adaptable system for interpreting natural language queries*". The two strings have the same semantic content, but show a different wording.

Although the human eye is easily able to identify a strong resemblance between them, a simple pattern matching algorithm encounters major difficulties. In addition, the size of the data must also be kept in mind. Considering the complexity of the problem, it was decided to examine only citations inside the corpus, therefore each title extracted from a reference has been compared with each title in the corpus metadata. Although there are several algorithms to compare strings, most of them are not scalable with large amounts of data. We used the *SequenceMatcher* class provided by the *difflib* library which implements a variant of a pattern matching algorithm published in 1980 by Ratcliff and Obershelp [16]. It is based on the principle of finding the longest common sequence, free of undesired elements (called junk). SequenceMatcher also supports heuristics for the automatic identification of junk elements. This process takes place calculating how many times each element appears in the sequence. If the duplicates of an element (after the first one) represent more than 1% of the sequence and the sequence is at least 200 elements long, then the element is marked as "popular" and is considered as junk (the version proposed by Ratcliff and Obershelp did not foresee the existence of junk elements). The search for common sequences is then recursively repeated, thus producing judicious correspondences.

The complexity of SequenceMatcher is quadratic in the worst case and linear in the best case, differently from the original version (as proposed by Ratcliff-Obershelp) that presented a cubic complexity in the worst case and quadratic in the best case. Given the complexity of the calculation, *difflib* provides three variants to the calculation of similarity (ratio, quick-ratio, real-quick-ratio) which gradually return an approximation of the exact value. In our case we used *quick-ratio* which represents a good compromise between correctness and efficiency.

The risk of having few associations among articles due to a too high degree of requested similarity carried to the choice of a reasonable threshold. This was finally set to 92%, when, in 100 random matches identified, none turned out to be a false positive. In this way, the risk of associating different articles has been reduced.

## 4 Semantic Analysis

### 4.1 Text Transformation

As usually done when dealing with textual documents, it was necessary to convert them into vector-based distributional representations. We first created a dictionary containing the words contained in the documents (taken only once). Then, for each document $i$, we counted the number of occurrences of each word $w$. The value is stored in an array in position $X$ *[i, j]*, where $j$ is the $w$-index in the dictionary. For this operation, we used the implementation provided by the *scikit-learn* library[4].

The implementation of the vector transformation algorithm also includes the possibility of using n-grams instead of single words, filtering out the stopwords and customizing the dictionary creation function (using the *tokenizer* function which splits the text into tokens, later used as features). However, occurrences are known to be improvable numeric representations of a corpus since they are not normalized with respect to documents length and spreading of the words over the document collection. To avoid the problem, we employed the well-known Term Frequency - Inverse Document Frequency (tf-idf) weighting strategy.

### 4.2 Syntax-based Snippet Tokenization

The terms are provided by the tokenizer function which splits a string into tokens composed of single words or, in our case, groups of words (n-grams). The choice of the tokens is particularly crucial in the proposed method, since one of the main objectives of the final graph is to differentiate the citations according to their intention, or meaning.

The proposed approach focusses on finding those words which are syntactically linked to the citation (dubbed *trigger words* from now on). To explain in detail the method, consider the following example: the sentence extracted from the snippet '*X compared the difference between citation terms extracted*', where $X$ takes the place of the reference. The syntactic parser [14] returns the syntactic tree depicted in Figure 1.

The first step is the identification of the words which are directly related to the citation, without taking into consideration the orientation and the type

[4] `http://scikit-learn.org/stable/`

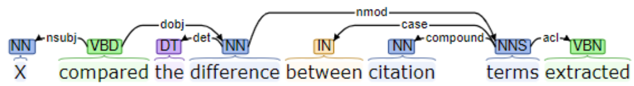



**Fig. 1.** Example of syntatic tree.

of the syntactic dependencies. Following the example, the tree visit begins with *compared*". Afterwards, the process continues by only considering a specific set of dependencies[5] and following the orientation of the graph, up to a maximum depth of $d = 2$. In the example, the words extracted were "*compared*", "*difference*", and "*terms*".

Finally, since single words can carry little information, we proceeded to create n-grams by composing the trigger-words with all the words that separate them from the reference. In the example, we get:

- *compared*
- *compared the difference*
- *compared the difference between citation terms*

In order to generalize the obtaned tokens, and therefore avoiding an excessively sparse matrix, a stemming[6] algorithm has been applied to the strings. By applying this on the previous example we obtain:

- *compar*
- *compar the differ*
- *compar the differ between citat term*

These three n-grams represent the tokens generated by our syntax-based tokenizer on the input snippet, and therefore the base units on witch the tf-idf computation works on. A maximum limit of 10K features (i.e. tokens generated as described above) has been set, taking the most frequent features only.

### 4.3 Dimensionality Reduction

It is often advisable to reduce the number of dimensions of a matrix before it is analyzed by a clustering algorithm. The reduction of the components is in fact part of the standard procedure applied to the study of large datasets and avoids numerous side effects (e.g., curse of dimensionality) that would compromise the results. In the specific case concerning our proposal, the matrix size was originally around 360K (snippets) x 10K (features). However, we reduced the number of snippets by 39% by removing 0-valued vectors, obtaining a final matrix of 219K x 10K. Then, to reduce the number of dimensions, we used the well-known Latent Semantic Analysis [4], using the implementation provided by *scikit-learn*.

[5] *nsubj*, *csubj*, *nmod*, *advcl*, *dobj*.

[6] The Porter Stemmer has been adopted.

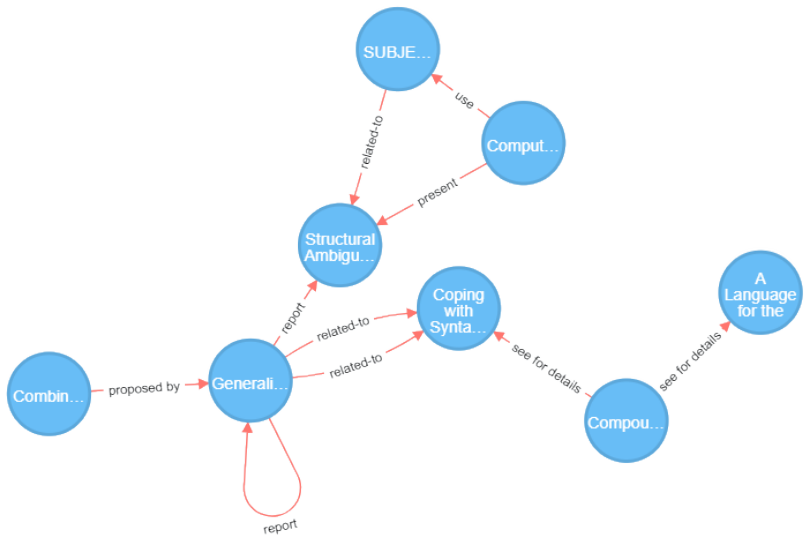



**Fig. 2.** Example of the citation graph of a publication: nodes represent publications, while edges express citations. Edge labels show extracted citation meanings.

## 5 Citation Meanings Extraction

### 5.1 Citation Clustering

For data with a large number of dimensions, such as text documents represented by tf-idf scores, the cosine similarity turned out to be a better metric than the Euclidean distance, as usual when dealing with textual data. We used KMeans in its minibatch variant with samples of 1000 elements. This way, the algorithm converges faster than the other tested methods (DBSCAN, hierarchical clustering) and produces results comparable, by quality, to the canonical version of KMeans. In addition, for this purpose, we made use of the initialization *k-means++* provided by *scikit-learn* which initializes the centroids maximizing the distance. Following numerous experiments, and considering the results provided by the analysis of the clustering silhouette, the value of K was set to 30. Figure 3 shows a small portion of the constructed graph.

The function provided by *scikit-learn* returns the spatial coordinates of the centroids, in the original data space (i.e., not reduced), and the cluster to which each sample belongs. Thanks to this information it was possible to obtain a top-terms ranking for each cluster, i.e., the features which were closer to the centroids and therefore more representative of that cluster.

Finally, starting from the clustering, the silhouette coefficient was calculated for each cluster. Since the implementation of *scikit-learn* does not provide the possibility of sampling the data and considering the large number of existing

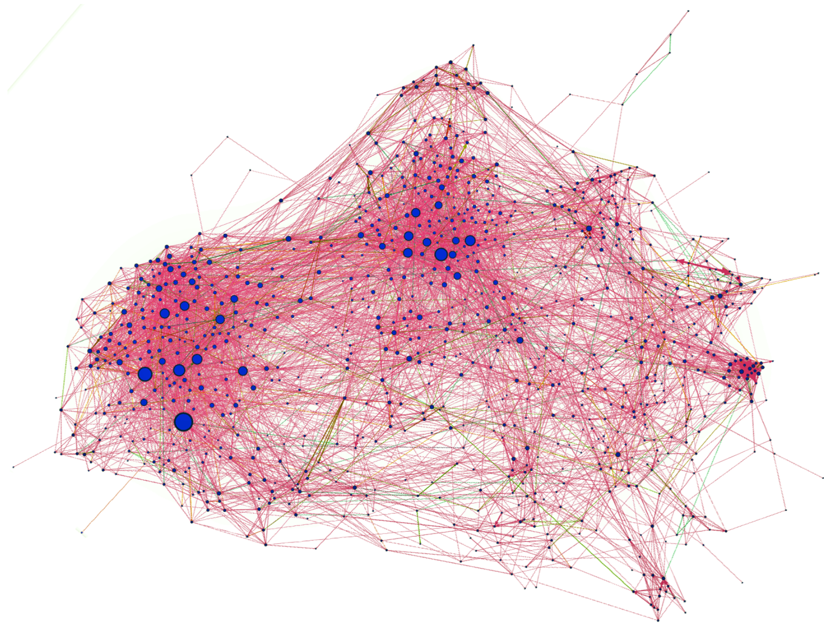
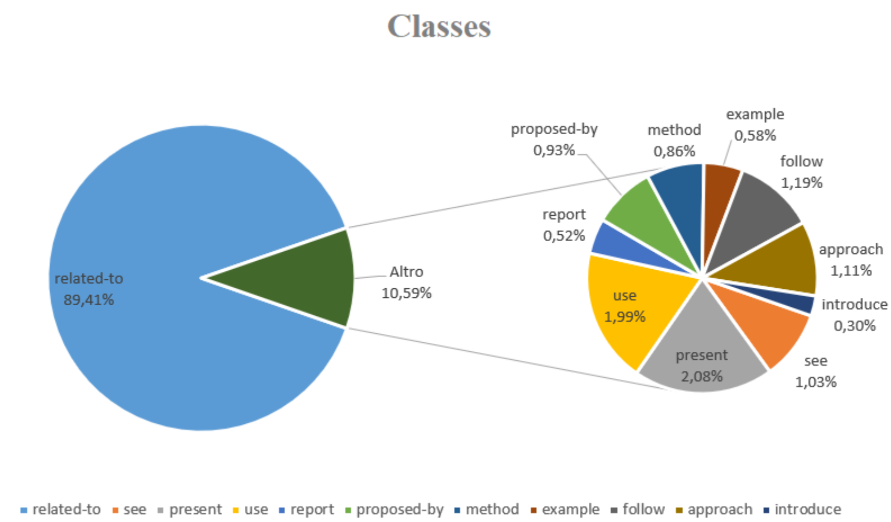


**Fig. 3.** Small portion of the graph, containing 947 nodes (4,5%) and 6267 relations (6,5%), visualized by the software Gephi, filtering the nodes according to their degree and highlighting, with different colors, the different classes of citations.

samples, it was necessary to implement our own version using a random sampling of 900 samples for each cluster. The results were then shown in a graph showing the average of the coefficient by means of a red line. Then, from the calculation of the silhouette, the cluster with the highest number of elements was excluded. In fact, it contained elements which were not semantically coherent with each other. The clustering phase ends by also producing an XML file containing the snippets (identified by an alphanumeric identifier) divided by associated cluster.

### 5.2 Cluster Labeling

The information gathered from the clustering phase has been subsequently analyzed and the classes corresponding to the clusters of particular interest have been assigned manually. The decision was taken by taking into account all the aspects highlighted by the clustering, in particular the silhouette values, the cardinality of each clusters, the top-terms and samples extracted from the clusters. Clusters that capture a particular aspect were labeled by storing this information in a *.json* file has been then used for the construction of the graph (see Figure 2 for an example).




| related-to | see | present | use | report | proposed-by | method | example | follow | approach | introduce |
|---|---|---|---|---|---|---|---|---|---|---|
| 85428 | 984 | 1986 | 1906 | 501 | 886 | 819 | 557 | 1140 | 1060 | 284 |

**Fig. 4.** Citation meanings (or classes) and relative number of extracted citations.

The largest cluster has been discarded a priori. In fact, alone, it represents the 67% of the data and contains very heterogeneous information, such as *('show')*, *('task')*, *('propos', 'in')* *('work')*, *('translat')*, *('word')*, etc. Some other clusters have been rejected for the same reason, showing also very low silhouette values. All these groups of citations have been labeled with the generic label *related-to*.

At the end of the selection process, 9 clusters (also called *citation meanings* from now on) were preserved and manually associated with labels. For example, one cluster having the top-terms: *('see')*, *('for', 'detail')*, *('see', 'X')*, *('for', 'an', 'overview')*, *('for', 'more', 'detail')*, *('see', 'X', 'and')*, *('detail', 'see')*, *('for', 'discussed')*, *(' for ',' a ',' discussed ')*, *(' for ',' further ',' detail ')*, has been associated with the label: *see-for-details* as it captures a particular type of citation aimed at providing further details on the subject dealt with in the article.
Examples of extracted snippets from the cluster are:

- *For a good discussion of the differences, see [X]*
- *See [X] for an overview and history of MUC6 and the 'Named Entity*
- *Actually, the extensions concern the use of tries (see [X]) as the sole storage device for all sorts of lexical information in MONA*
- *In (1) SMES is embedded in the COSMA system, a German language server for existing appointment scheduling agent systems (see [X], this volume, for more information).*

**Table 1.** Considered labels and relative meanings. $A$ is the paper that mentions the paper $B$, while $c$ represents something extracted from the snippet which is only relative to $B$ (in that case, $A$ has the only role of containing the relationship between $B$ and $c$). Even if we decided to keep the last three labels separated, they can be considered as depicting the same semantic relation.

| **Label** | **Type** | **Meaning** |
|---|---|---|
| see-for-details | $A \longrightarrow B$ | See $B$ for more details |
| follow | $A \longrightarrow B$ | The authors of $A$ followed what done in $B$ |
| method | $B \longrightarrow c$ | $B$ uses the method $c$ |
| approach | $B \longrightarrow c$ | $B$ follows the approach $c$ |
| use | $B \longrightarrow c$ | $B$ makes use of $c$ |
| report | $B \longrightarrow c$ | $B$ reports $c$ (usually results) |
| present | $B \longrightarrow c$ | $B$ presents $c$ |
| proposed-by | $B \longrightarrow c$ | $B$ proposes $c$ |
| introduce | $B \longrightarrow c$ | $B$ introduces $c$ |

The entire set of identified labels are shown in Table 1, quantitatively distributed as in Figure 4. Among the 9 identified citation meanings, we found that they belong to two different types:

***A*** $\longrightarrow$ ***B*** - In this case, article A cites article B expressing some semantics about A and B, directly. This happens for labels *see-for-details* and *follow*.

***B*** $\longrightarrow$ ***c*** - Citations of this type are contained in A, but express some semantics about article B only. The term $c$ may refer to different concepts (methods, algorithms, results, etc.). The labels of this type are *method*, *approach*, *use*, *report*, *present*, *proposed-by*, and *introduce*.

It is important to note that in the latter case our method is able to extract (and use, for exploration purposes) semantic information about articles once cited in third-party articles. To the best of our knowledge, this is one of the first work following this approach. Differently from extracting information about papers directly within them, by doing it where they are referenced, there is some certainty about the recognized relevance of the extracted knowledge. Always in the $B \longrightarrow c$ case, we changed the $A \longrightarrow B$ label to *highly-related-to*.

A graph of 21,148 nodes (corresponding to the articles of the corpus) and 95,551 edges (representing the citations) was finally created. It has been observed that 8,416 nodes (around 40%) do not participate in any relation because they do not have citations to articles inside the corpus.

## 6 Model and Method Complexity

Starting from the snippets extracted in the preprocessing phase, 360,162 sentences were obtained, being then analyzed syntactically, according to the described method, creating a document-term matrix of 360,162 x 10,000. The operation required 4 hours of processing on a computer equipped with an Intel

Core i7 6500U, 12 GB of RAM, 1 TB HDD and 128GB SSD. From the tf-idf matrix, 140,872 lines (39% of the total) were removed, composed exclusively of null values. The final matrix (219,290 x 10,000) turned out to be more manageable by the reduction and clustering algorithms. The reduction through the truncatedSVD algorithm required the most memory consumption, quickly saturating the memory available from the machine and, for this reason, it took 84 minutes to produce the results. The 3,000-component reduction has preserved 83% of the variance. Thanks to the use of the minibatch variant, KMeans finished in just 15 seconds producing the 30 required clusters.

## 7 Evaluation

### 7.1 Application: Citation Explorer

In order to assess the effectiveness of the proposed method, we have developed a simple web application that allows the navigation of the literature through the extracted semantically-enriched citation paths. An online version is available at `http://citexp.di.unito.it`.

The collected and assembled data have been then stored in two csv files, containing the nodes and the edges of the graph respectively. A header is added to the csv document that specifies the structure. In this way, the elaboration of the ACL corpus ends, producing a result that is independent of the technology that can be used to visualize and analyze it. In our application, the data have been uploaded to a graph database (Neo4j[7]) on which basic queries were carried out to obtain statistics and validate the project objectives. The graph is constructed by composing information derived from the corpus metadata, the composition of the clusters, and the assigned labels. In the graph each node represents an article and stores information such as title, authors and, where present, the date of publication. The edges that connect the nodes represent the citations, characterized by the class (or label) and therefore by the type of reference. Unlabeled arches have been preserved, labeled with a generic *related-to* relation. Starting from a specific node it is in fact possible to navigate the graph following the path provided by the edges and filtering the nodes based on the information memorized by the database: title, authors and date of the article, class and section to which the reference belongs (see Figure 5 for an example of interaction.).

In the application, the result of a standard search for papers includes several articles ordered by the weight (i.e, the rank) of the articles calculated with the PageRank algorithm [21]. Once an article of interest has been identified, the article details are shown together with a list of relevant articles filtered according to the citation classes. The selection algorithm is also based on PageRank and provides, in order of weight, both articles cited by the selected article and those that mention it. The application also includes several other interaction features such as tooltips showing the sections and the sentences from which the citations were taken, the trend of citations over the years, and so forth. The trend is

[7] `https://neo4j.com`

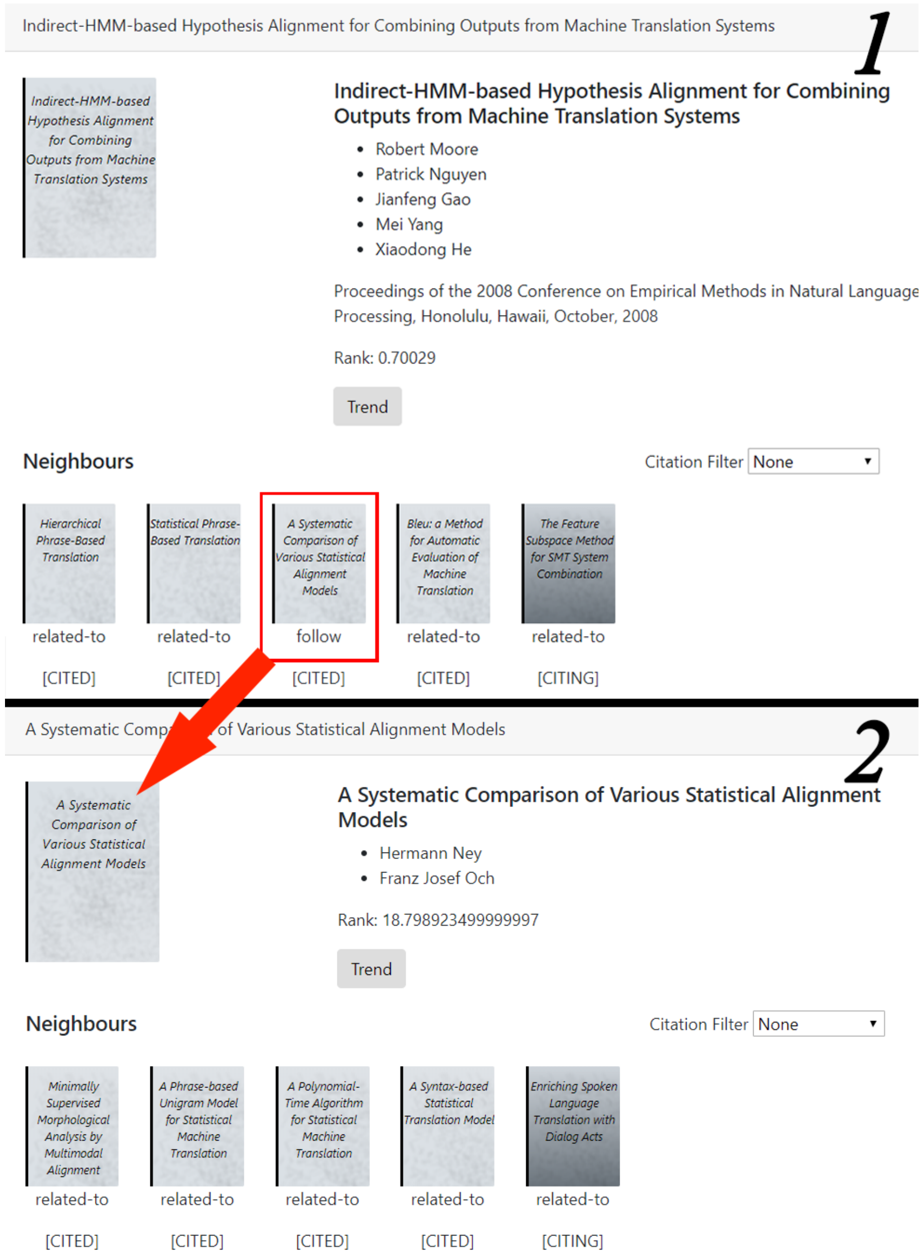



**Fig. 5.** Example of visualization and interaction (P1 *follow* $\longrightarrow$ P2) with the developed web-based application, available at `http://citexp.di.unito.it`.

represented by the number of citations referring the article over the years and can be used, for example, to identify emerging articles or articles representing a foundation for ongoing research.

### 7.2 Reliability of the Extracted Citation Meanings

To give an overview of the reliability of the approach, we produced a validation dataset containing all citation texts with the automatically-associated citation meanings. In detail, we manually evaluated the correctness of 100 random instances for 9 citation meanings (i.e., 900 snippets)[8]. Table 2 reports the result of this evaluation. The resulting overall accuracy is 95.22%, demonstrating the high efficacy of the method in associating correct meanings to the citations[9].

**Table 2.** Accuracy of the manual validation. A set of 100 random snippets (for each extracted citation meaning) have been manually checked for correctness.

| | *see-for-details* | *proposed-by* | *introduce* |
|---|---|---|---|
| **Accuracy** | 98% | 98% | 93% |
| | *follow* | *approach* | *method* |
| **Accuracy** | 96% | 89% | 96% |
| | *report* | *present* | *use* |
| **Accuracy** | 97% | 98% | 92% |

## 8 Conclusions and Future Work

We have introduced a new method for an advanced search and exploration of a large body of scientific literature, presenting a methodology comprising a pipeline and a set of tools for structuring and labeling citations.[10] Specifically, the proposed approach is based on a Natural Language Processing architecture and a semi-supervised clustering phase. To demonstrate the validity of the proposal, we first manually evaluated a random selection of the extracted knowledge on a large collection of scientific papers in the Computational Linguistics field (ACL Anthology). Then, we developed a freely-accessible web-based application, which will be maintained and further developed along the years for research purposes). In particular, through the automatically-extracted citation meanings, it can be possible to browse the literature by following fine-grained types of citations, thus

[8] We excluded from the evaluation the 10th cluster *related-to* since it included the remaining citations having a very broad scope

[9] Since we did not have a complete labeled corpus with positive and negative examples, we could not compute standard Precision/Recall/F-measures.

[10] Both documentation and source code of the pipeline, as well as the complete set of citation snippets per category and the graph, are available at `https://github.com/rogerferrod/citexp`

providing an enhanced retrieval process, with better at-a-glance overviews over the state of the art. In future work, we aim at applying the method on a broader domain, and at a larger scale.

A set of open issues emerged from this work. For example, in relation with the decomposition of the text in sentences, more effort should be spent since the used algorithms are not specific for scientific texts.

Finally, since the project dealt exclusively with the research of a methodology for the organization and storage of the scientific literature, it is reasonable to include tools to query the database for non-specialized users. Solutions of this kind include browser plugins, web platforms able to present the articles according to a narration (storytelling) dictated by the temporal evolution of the topics, or instruments related to bibliometric analyses of the articles.

## References

1. Akujuobi, U., Zhang, X.: Delve: A dataset-driven scholarly search and analysis system. SIGKDD Explor. Newsl. **19**(2), 36–46 (Nov 2017). https://doi.org/10.1145/3166054.3166059, `http://doi.acm.org/10.1145/3166054.3166059`
2. Alexander, E., Kohlmann, J., Valenza, R., Witmore, M., Gleicher, M.: Serendip: Topic model-driven visual exploration of text corpora. In: Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on. pp. 173–182. IEEE (2014)
3. Bergstrm, P., Atkinson, D.C.: Augmenting the exploration of digital libraries with web-based visualizations. In: 2009 Fourth International Conference on Digital Information Management. pp. 1–7 (Nov 2009). https://doi.org/10.1109/ICDIM.2009.5356798
4. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6), 391–407 (1990)
5. Diederich, J., Balke, W.T., Thaden, U.: Demonstrating the semantic growbag: Automatically creating topic facets for faceteddblp. In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. pp. 505–505. JCDL '07, ACM, New York, NY, USA (2007). https://doi.org/10.1145/1255175.1255305, `http://doi.acm.org/10.1145/1255175.1255305`
6. ubelj, Nees Jan van Eck, Ludo Waltman, L.: Clustering Scientific Publications Based on Citation Relations: A Systematic Comparison of Different Methods. PLoS ONE 11(4) (2016)
7. van Eck, Ludo Waltman, N.J.: VOS: a new method for visualizing similarities between objects. Advances in Data Analysis: Proceedings of the 30th Annual Conference of the German Classification Society (pp. 299-306). Springer (2007)
8. van Eck, Ludo Waltman, N.J.: CitNetExplorer: A new software tool for analyzing and visualizing citation networks. Journal of Informetrics, 8(4), 802-823 (2014)
9. van Eck, Ludo Waltman, N.J.: Citation-based clustering of publications using CitNetExplorer and VOSviewer. Scientometrics, Volume 111, Issue 2, pp 10531070 (2017)
10. Kan, I.G.C.C.L.G.M.Y.: ParsCit: An open-source CRF reference string parsing package. in Proceedings of the Language Resources and Evaluation Conference (LREC 08), Marrakesh, Morrocco, May (2008)

11. Kataria, S., Mitra, P., Bhatia, S.: Utilizing context in generative bayesian models for linked corpus. In: AAAI. vol. 10, p. 1 (2010)
12. Kim, J., Kim, D., Oh, A.: Joint modeling of topics, citations, and topical authority in academic corpora. arXiv preprint arXiv:1706.00593 (2017)
13. Li, H., Councill, I.G., Lee, W.C., Giles, C.L.: Citeseerx: an architecture and web service design for an academic document search engine. In: WWW (2006)
14. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014), `http://www.aclweb.org/anthology/P/P14/P14-5010`
15. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 169–178. ACM (2000)
16. Metzener, J.R.D.: Pattern Matching: The Gestalt Approach. Dr. Dobbs Journal (1988)
17. Mutschke, P.: Mining networks and central entities in digital libraries. a graph theoretic approach applied to co-author networks. In: International Symposium on Intelligent Data Analysis. pp. 155–166. Springer (2003)
18. Nagwani, N.: Summarizing large text collection using topic modeling and clustering based on mapreduce framework. Journal of Big Data **2**(1), 6 (2015)
19. Newman, M.E.: Scientific collaboration networks. i. network construction and fundamental results. Physical review E **64**(1), 016131 (2001)
20. Oelke, D., Strobelt, H., Rohrdantz, C., Gurevych, I., Deussen, O.: Comparative exploration of document collections: a visual analytics approach. In: Computer Graphics Forum. vol. 33, pp. 201–210. Wiley Online Library (2014)
21. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab (November 1999), `http://ilpubs.stanford.edu:8090/422/`, previous number = SIDL-WP-1999-0120
22. Popescul, A., Ungar, L.H., Flake, G.W., Lawrence, S., Giles, C.L.: Clustering and identifying temporal trends in document databases. In: adl. p. 173. IEEE (2000)
23. Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P., Steyvers, M.: Learning author-topic models from text corpora. ACM Transactions on Information Systems (TOIS) **28**(1), 4 (2010)
24. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: Proceedings of the 20th conference on Uncertainty in artificial intelligence. pp. 487–494. AUAI Press (2004)
25. Shotton, S.P.D.: FaBiO and CiTO: ontologies for describing bibliographic resources and citations. Web Semantics: Science, Services and Agents on the World Wide Web. Volume 17, Pages 33-43 (2012)
26. Strapparava, R.M.C.C.C.: Corpus-based and knowledge-based measures of text semantic similarity. AAAI'06 Proceedings of the 21st national conference on Artificial intelligence, Volume 1, Pages 775-780 (2006)
27. Tu, Y., Johri, N., Roth, D., Hockenmaier, J.: Citation author topic model in expert search. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 1265–1273. Association for Computational Linguistics (2010)