Analysis of lexical semantic changes in corpora with the Diachronic Engine

Pierluigi Cassotti and **Pierpaolo Basile** and **Marco de Gemmis** and **Giovanni Semeraro**
Department of Computer Science
University of Bari Aldo Moro
Bari, Italy
{firstname.surname}@uniba.it

CEUR-WS.org/Vol-2769/paper_71.pdf

Abstract

English. With the growing availability of digitized diachronic corpora, the need for tools capable of taking into account the diachronic component of corpora becomes ever more pressing. Recent works on diachronic embeddings show that computational approaches to the diachronic analysis of language seem to be promising, but they are not user friendly for people without a technical background. This paper presents the *Diachronic Engine*, a system for the diachronic analysis of corpora lexical features. *Diachronic Engine* computes word frequency, concordances and collocations taking into account the temporal dimension. It is also able to compute temporal word embeddings and time-series that can be exploited for lexical semantic change detection.

1 Motivation and Background

Synchronic corpora are widely used in linguistics for deriving a set of abstract rules that govern a particular language under analysis by using statistical approaches. The same methodology can be adopted for analyzing the evolution of word meanings over time in the case of diachronic corpora. However, this process can be very time-consuming. Usually, linguists rely on software tools that can easily explore and clean the corpus, while highlighting the more relevant linguistic features. Sketch Engine[1](Kilgarriff et al., 2004; Kilgarriff et al., 2014) is the leading tool in the corpus analysis field. Beyond several interesting features, Sketch Engine includes *trends* (Kilgarriff et al., 2015), which allow for diachronic analysis based on the frequency distribution of words. Trends rely on merely frequency features, ignoring word usage information. Moreover, the Sketch Engine interface does not provide temporal information about concordances and collocations. NoSketchEngine[2] is an open-source version of SketchEngine. It requires technical expertise for the setup and, contrarily to SketchEngine, it does not support word sketches, terminology, thesaurus, n-grams, trends and corpus building. An interesting system is DiaCollo[3] (Jurish and der Wissenschaften, 2015), a software tool for the discovery, comparison, and interactive visualization of target word combinations. Combinations can be requested for a particular time period, or for a direct comparison between different time periods. However, DiaCollo is focused exclusively on the extraction and visualization of collocations from diachronic corpora.

In recent works about computational diachronic linguistics, techniques based on word embeddings produce promising results. In Semeval Task 1 (Schlechtweg et al., 2020), for instance, type embeddings rich high performances on both subtasks. However, these techniques are not included in any aforementioned linguistic tool. In order to bridge this gap, we try to build a tool that includes approaches for the analysis of diachronic embeddings. The result of our work is Diachronic Engine (DE), an engine for the management of diachronic corpora that provides tools for change detection of lexical semantics from a frequentist perspective. DE includes tools for extracting diachronic collocations, concordances in different time periods as well as for computing semantic change time-series by exploiting both word frequencies and word embeddings similarity over time.

The rest of the paper is organized as follows:

Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]https://www.sketchengine.eu/

[2]https://nlp.fi.muni.cz/trac/noske

[3]https://www.clarin.eu/showcase/diacollo

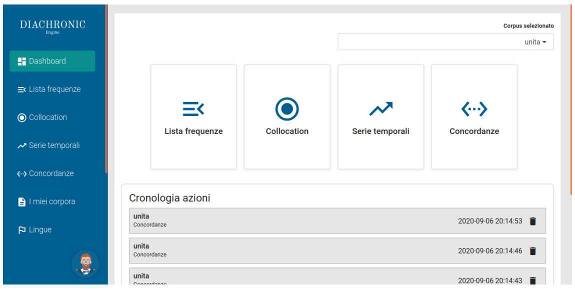

Figure 1: Diachronic Engine web interface.

Section 2 describes the technical details of DE, while Section 3 shows some use cases of our engine that encompass that address time-series. We also present the results of a preliminary evaluation about the system's usability in Section 4. Conclusions and future work close the paper.

2 Diachronic Engine

Diachronic Engine (DE) is a web application for lexical semantic change analysis in diachronic corpora. The DE pipeline needs diachronic corpora to compute statistics about the corpus. A diachronic corpus must include a temporal feature (e.g., year or timestamp of the publication date); DE exploits that feature to sort the documents.

We adopt the vertical format to represent word information, as specified for the IMS Corpus Workbench (CWB). In a vertical corpus, each word is in a new line. In each line, fields, called p-attributes, are separated by tabs. In DE the default p-attributes are word, lemma, PoS tag and syntactic dependency. Non-recursive XML tags (s-attributes) on a separate line can be used for representing sentences, paragraphs and documents.

Corpora can be served in vertical format[4] or in plain-text mode; in the latter case, the plain-text is transformed in vertical format using the Spacy UDPipe[5] (Straka, 2018) tool, which splits plain-text into sentences and then predicts the PoS-tag, the lemma and the syntactic dependency for each token. UDPipe is a dependency parser that provides models for several languages. Models are built by using the Universal Dependencies [6] datasets as training data. Input files' names must contain the temporal tag of the period to which they refer. DE automatically detects temporal patterns in the name of the files. In particular, the last sequence of numbers in the file name is used to sort the documents.

Corpora are stored and managed by the CWB, a tool for the manipulation of large, linguistically annotated corpora. In particular, DE relies on the Corpus Query Processor (CQP) (Christ et al., 1999), a specialized search engine for linguistic research.

For building temporal word embeddings, DE exploits Temporal Random Indexing (TRI) (Basile et al., 2014; Basile et al., 2016) that computes a word vector for each time period by summing shared random vectors over all the periods. TRI is able to produce aligned word embeddings in a single step and it is based on Random Indexing (Sahlgren, 2005), where a word vector (word embedding) $sv_j^{T_k}$ for the word w_j at time T_k is the sum of random vectors r_i assigned to the co-occurring words taking into account only documents $d_l \in T_k$. Co-occurring words are defined as the set of m words that precede and follow the word w_j. Random vectors are vectors initialized randomly and shared across all time slices so that word spaces are comparable.

Future versions will include other approaches, such as Procustes (Hamilton et al., 2016), Dynamic Word Embeddings (Yao et al., 2018), Dynamic Bernoulli Embeddings (Rudolph and Blei, 2018) and Temporal Referencing (Dubossarsky et al., 2019).

The DE architecture is based on the client-server paradigm. The back-end of DE has been developed with Flask, a web framework written in Python. Concordances are retrieved by CQP, that indexes the corpus as soon as it is uploaded to the server, while collocations and frequencies are computed in Python. The back-end provides a set of services by a REST API where the input/output is based on JSON messages.

The back-end consists of three macro components: User Handler, Corpus Handler and Diachronic Operations. The User Handler manages registered users information such as username and passwords. Admitted operations on users are creation, read, update and delete. The Corpus Handler Component manages corpora information such as name, language, the list of fields in the vertical files, corpus visibility. Moreover, it deals with corpora types: each corpus has a label indicating if it is synchronic or diachronic. For di-

[4] https://www.sketchengine.eu/my_keywords/vertical/
[5] https://pypi.org/project/spacy-udpipe/
[6] http://universaldependencies.org

achronic corpora also the temporal range is stored. Operations admitted on corpora are: creation, update, delete, search and read. The Diachronic Operations component shows frequency lists, collocations of words, time-series, change-points and concordances. This component relies on CWB that indexes vertical files.

The Diachronic Operations component architecture is sketched in Figure 2.

The front-end of DE has been developed with JHipster[7], using Spring[8] for server-side applications and Angular for client-side applications. The front-end communicates with the back-end by the means of the REST API.

The front-end design is inspired by the Google's Material Design and the Sketch Engine interface. The user interface provides multilingual support in Italian and English, but we plan to extend it to other languages.

This architecture allows the independence between the back-end and the front-end, in this way is possible to develop a different front-end or connect the front-end to a different implementation of the back-end. The only constraint is the REST API interface.

A screenshot of the DE homepage is provided in Figure 1. The homepage provides an easy access to all corpora owned by the logged user with links to available tools. The front-end provides also tools for creating and managing users and corpora. In particular, it is possible to define different grant permissions for each corpus.

The tool is distributed as open-source software under the GNU v3 license[9].

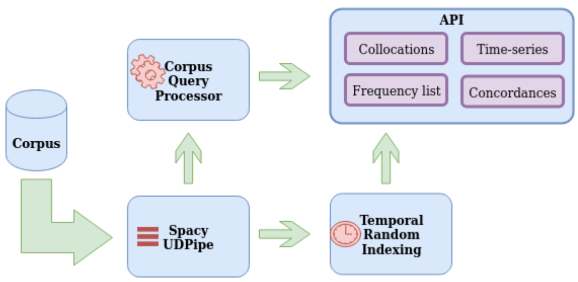

Figure 2: Diachronic Engine corpora manager.

[7]https://www.jhipster.tech/
[8]https://spring.io/
[9]https://github.com/swapUniba/Diachronic-Engine

2.1 DE tools

DE provides a set of tools for managing and querying diachronic corpora. The core of the back-end is based on the IMS Open Corpus Workbench (CWB) [10], which allows querying the indexed corpora by using the powerful CQP. Other tools have been integrated to facilitate the analysis of a diachronic corpus:

Word frequency Many works show a correlation between lexical semantic change and frequency differences between time periods. Google Ngram Viewer (Michel et al., 2011) uses n-grams frequencies over time to show the change in the semantics of n-grams. SketchEngine exposes the Trends tool, which uses a linear regression of frequencies to predict words that appear to be changed. In DE, queries can be filtered by part-of-speech, as well as by time periods. We use normalized frequencies, that can be filtered by time period.

Collocations Collocations have shown to be an effective tool in diachronic analysis (Basile et al., 2019). A collocation is a sequence of words that occurs more often than would be expected. In order to compute the collocation strength we use the logDice (Rychlỳ, 2008):

$$log \frac{2f_{xy}}{f_x + f_y}$$

logDice takes into account the frequency of the word f_x, of the collocate f_y and the frequency of the whole collocation f_{xy}. Collocation results can be grouped by the PoS tag.

Concordances Concordances offer a way to find "the evidence" directly in the text by exploiting the context. The Concordances tool lists instances of a word with its immediate left and right context and the period the collocation belongs to. An example of concordances from "L'Unità" (Basile et al., 2020), is shown in Figure 3.

Time-series A time-series $\Gamma(w)$ of a word w is an ordered sequence of cosine similarities between the word vector at time k (v_w^k) and the previous one at time $k-1$ (v_w^{k-1}):

$$\Gamma(w)_k = \frac{v_w^k \cdot v_w^{k-1}}{|v_w^k||v_w^{k-1}|}$$

[10]http://cwb.sourceforge.net/

#	Source	Date	Left context	KWIC	Right context	Copy
1	unita	1948-01-01	Forze Aeree Israelite L aereo	pilotato	da ufficiali ebrei era diretto	
2	unita	1951-01-01	casa , su tm aereo	pilotato	da lo stesso comandante e	
3	unita	1951-01-01	l apparecchio , che era	pilotato	da il tenente Augusto Sb^rtoli	

#	Source	Date	Left context	KWIC	Right context	Copy
581	unita	2008-01-01	. Il presidente che ha	pilotato	gli Usaversodue conflitti da gli	
582	unita	2009-01-01	aveva parlato di « incidente	pilotato	e programmato » , a	
583	unita	2009-01-01	di Milano , che avrebbe	pilotato	un' asta giudiziaria per assegnare	

Figure 3: DE shows the KWIC (Keyword in the context) "pilotato", shifted from meaning *driven* to meaning *manipulated*.

Diachronic Engine relies on word vectors computed by Temporal Random Indexing, but it is possible to integrate other approaches. In order to detect change points, we use the Mean Shift algorithm (Taylor, 2000). According to this model, we define a mean shift of a general time series Γ pivoted at time period j as:

$$K(\Gamma) = \frac{1}{l-j} \sum_{k=j+1}^{l} \Gamma_k - \frac{1}{j} \sum_{k=1}^{j} \Gamma_k \quad (1)$$

In order to understand if a mean shift is statistically significant at time j, a bootstrapping (Efron and Tibshirani, 1994) approach under the null hypothesis that there is no change in the mean is adopted. In particular, statistical significance is computed by first constructing B bootstrap samples by permuting $\Gamma(t_i)$. Second, for each bootstrap sample P, $K(P)$ is calculated to provide its corresponding bootstrap statistic and statistical significance (p-value) of observing the mean shift at time j compared to the null distribution. Finally, we estimate the change point by considering the time point j with the minimum p-value score. The output of this process is a ranking of words that potentially have changed meaning. Time-series is able to compare multiple words at the same time and allows to filter words by time period.

3 Use cases

In this section, we describe two use cases concerning both historical and computational linguistics. DE is an extension of existing tools for synchronic corpora. It shares many of the use cases already available on those tools, such as applications in lexicography, terminology and linguistics.

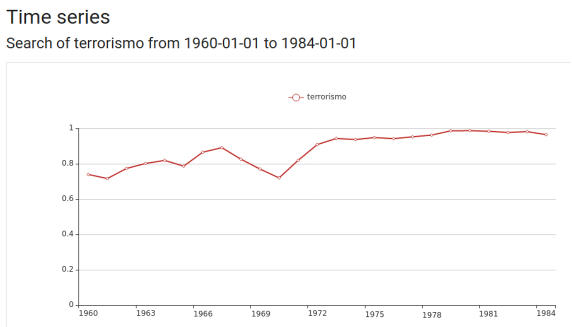

Figure 4: DE shows time-series of the word "terrorismo".

3.1 Event detection through time-series

Lexical semantic changes can reveal aspects of real-world events, such us global armed conflicts (Kutuzov et al., 2017). DE provides several tools to help events detection through time-series:

- the comparison of two time-series for highlighting potential correlations between lexical-semantic changes
- the plot of the time-series of cosine similarity between two word vectors over time, showing how the relatedness between two words changes over time
- the detected change points can bring out hidden information

In Figure 4, the time-series of "terrorismo" (*terrorism*) is shown. The time-series appears to be influenced by real-world events happening in Italy. In particular, we can observe a decrease in similarity starting in 1968 and culminating in 1970 during a crucial moment in Italy: "Anni di piombo" (*Years of Lead*), years marked by terrorism and violent clashes carried out by political activists.

3.2 Annotation of semantic shifts

The manual annotation of lexical-semantic shifts can be very expensive. Although robust frameworks (Schlechtweg et al., 2018) for the annotations already exist and are successfully used in evaluation tasks (Schlechtweg et al., 2020), no tools for facilitating the annotation are available yet.

DE can provide useful tools for the annotation of semantic shifts:

1. Frequencies over time can be preliminary exploited to filter words that have good coverage in the years under analysis;
2. Change points in time-series offer an overall and intuitive idea of the potential semantic shifts;
3. Diachronic concordances and collocations can support the identification of the type of change (Blank, 2012), such as when a word gains or loses a meaning.

4 Evaluation

We place a particular focus on the usability of our tool by giving a satisfactory experience. To understand the strength and weakness of the user interface, we conduct a preliminary usability test, according to the eGLU protocol (Simone et al., 2015). We use 21 participants. As a first step of the evaluation, we want to test the system's usability by measuring the task success rate: the ratio of users able to accomplish a set of predefined tasks. We ask participants to perform four tasks and we compute the average task success over all the 21 participants. During the evaluation, all participants complete their tasks without difficulties except for the showing frequency list task, where they had some problems with the corpus selection. We have already fixed this issue: the user is warned to choose a corpus from those available if no corpus is selected.

Results of the evaluation are reported in Table 1.

Task	Avg. task success
User registration	1
Login and show user information	1
Add a corpus	1
Show frequency list	.8095
Overall	**.9523**

Table 1: Results of the usability evaluation.

Moreover, we designed and dispensed a questionnaire for measuring user satisfaction. The questionnaire is composed of ten questions about the usability and the design of DE with a Likert scale of five values. The questionnaire results return an average score of 84.05/100. The system appear likeable to use.

5 Conclusions

In this paper, we present the Diachronic Engine, a tool for the analysis of lexical semantic change. DE integrates and extends current tools for corpus analysis enabling the study of corpus diachronic features. DE includes tools not included in other systems, such as time-series and change points detection based on diachronic word embeddings.

As future work, we plan to provide pre-loaded corpora such as Google Ngram, Diacoris (Onelli et al., 2006) and the integration of other approaches for computing diachronic word embeddings. Moreover, we plan to add a tool for the annotation of lexical-semantic shifts inspired by DUREL (Schlechtweg et al., 2018).

Acknowledgments

The authors would like to thank Dr. Ferrante and Dr. Lopatriello for supporting the preliminary development of the Diachronic Engine (Ferrante, 2019; Lopatriello, 2020). This research has been partially funded by ADISU Puglia under the post-graduate programme "Emotional city: a location-aware sentiment analysis platform for mining citizen opinions and monitoring the perception of quality of life".

References

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. Analysing word meaning over time by exploiting temporal random indexing. In *First Italian Conference on Computational Linguistics CLiC-it (CLiC-it 2014)*. CEUR.org.

Pierpaolo Basile, Annalina Caputo, Roberta Luisi, and Giovanni Semeraro. 2016. Diachronic analysis of the Italian language exploiting google ngram. In *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016)*, page 56. CEUR.org.

Pierpaolo Basile, Giovanni Semeraro, and Annalina Caputo. 2019. Kronos-it: a dataset for the Italian semantic change detection task. In *Proceedings of the 6th Italian Conference on Computational Linguistics (CLiC-it 2019)*. CEUR.org.

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. A diachronic Italian corpus based on "L'Unità". In *Proceedings of the 7th Italian Conference on Computational Linguistics (CLiC-it 2020)*. CEUR.org.

Andreas Blank. 2012. *Prinzipien des lexikalischen Bedeutungswandels am Beispiel der romanischen Sprachen*, volume 285. Walter de Gruyter.

Oliver Christ, Bruno M Schulze, Anja Hofmann, and Esther Koenig. 1999. The ims corpus workbench: Corpus query processor (cqp): User's manual. *University of Stuttgart*, 8.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470. Association for Computational Linguistics (ACL).

Bradley Efron and RJ Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC Press.

Francesco Ferrante. 2019. Diachronic-engine: Un tool per la gestione dei corpora diacronici. B.Sc. degree Thesis in Metodi per il Ritrovamento dell'Informazione.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronie word embeddings reveal statistical laws of semantic change. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3, pages 1489–1501.

Bryan Jurish and Berlin-Brandenburgische Akademie der Wissenschaften. 2015. Diacollo: On the trail of diachronic collocations. In *Proceedings of the CLARIN Annual Conference*, pages 28–31.

Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. Itri-04-08 the sketch engine. *Information Technology*, 105:116.

Adam Kilgarriff, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlỳ, and Vít Suchomel. 2014. The sketch engine: ten years on. *Lexicography*, 1(1):7–36.

Adam Kilgarriff, Ondřej Herman, Jan Bušta, Vojtěch Kovář, et al. 2015. Diacran: a framework for diachronic analysis.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2017. Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*, pages 31–36.

Gabriele Lopatriello. 2020. Diachronic engine: A tool for the management of diachronic corpora. Master Thesis in Intelligent Information Access and Natural Language Processing.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Corinna Onelli, Domenico Proietti, Corrado Seidenari, and Fabio Tamburini. 2006. The diacoris project: a diachronic corpus of written Italian. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1212–1215.

Maja Rudolph and David Blei. 2018. Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference*, pages 1003–1011.

Pavel Rychlỳ. 2008. A lexicographer-friendly association score. *RASLAN 2008 Recent Advances in Slavonic Natural Language Processing*, page 6.

Magnus Sahlgren. 2005. An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering*.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic usage relatedness (durel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 169–174.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. Semeval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Borsci Simone, Boscarol Maurizio, Cornero Alessandra, et al. 2015. Il Protocollo eGLU 2.1. Il Protocollo eGLU-M. Come realizzare test di usabilità semplificati per i siti web ei servizi online delle PA. Glossario dell'usabilità.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Wayne A Taylor. 2000. Change-point analysis: a powerful new tool for detecting changes.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*, pages 673–681.