ClaimLinker: Linking Text to a Knowledge Graph of Fact-checked Claims

Evangelos Maliaroudakis
evanmaliar@gmail.com
Information Systems Laboratory,
ICS-FORTH & University of Crete
Heraklion, Greece

Katarina Boland
katarina.boland@gesis.org
GESIS, Cologne &
Heinrich-Heine-University
Düsseldorf
Cologne, Germany

Stefan Dietze
stefan.dietze@gesis.org
GESIS, Cologne &
Heinrich-Heine-University
Düsseldorf
Cologne, Germany

Konstantin Todorov
todorov@lirmm.fr
LIRMM, University of Montpellier,
CNRS
Montpellier, France

Yannis Tzitzikas
tzitzik@ics.forth.gr
Information Systems Laboratory,
ICS-FORTH & University of Crete
Heraklion, Greece

Pavlos Fafalios
fafalios@ics.forth.gr
Information Systems Laboratory,
ICS-FORTH
Heraklion, Greece

ABSTRACT

We present ClaimLinker, a Web service and API that links arbitrary text to a knowledge graph of fact-checked claims, offering a novel kind of *semantic annotation* of unstructured content. Given a text, ClaimLinker matches parts of it to fact-checked claims mined from popular fact-checking sites and integrated into a rich knowledge graph, thus allowing the further exploration of the linked claims and their associations. The application is based on a scalable, fully unsupervised and modular approach that does not require training or tuning and which can serve high quality results at real time (outperforming existing unsupervised methods). This allows its easy deployment for different contexts and application scenarios.

CCS CONCEPTS

• **Information systems → Content ranking**; **Web services**; **Web applications**; **Information systems applications**.

KEYWORDS

Claim Linking, Claim Retrieval, Knowledge Graphs, Fact-checking, Fake News

ACM Reference Format:
Evangelos Maliaroudakis, Katarina Boland, Stefan Dietze, Konstantin Todorov, Yannis Tzitzikas, and Pavlos Fafalios. 2021. ClaimLinker: Linking Text to a Knowledge Graph of Fact-checked Claims. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3442442.3458601

1 INTRODUCTION

Even though understanding the veracity of statements is important, in particular, when consuming information on the Web, fact-checking of arbitrary statements from Web documents on-the-fly is costly and impractical. On the other hand, the plethora of manually curated fact-checking portals such as Politifact, Snopes or FullFact have led to an abundance of information about claims and their corresponding truth ratings. However, such information is spread across a wide variety of portals. In addition, given that a particular claim proposition may occur in the form of diverse utterances [4], matching a given statement or utterance to fact-checked claims available from fact-checking portals remains a challenging problem.

This system demonstration presents ClaimLinker, a web application that builds on a knowledge graph of Web-mined fact-checked claims [12]. The system is composed of a set of modular components able to match arbitrary text to more than 30K claims and related data (such as truth ratings, authors, sources, entities and topics) mined from eight popular fact-checking sites and integrated into a knowledge graph. The application thus combines a semantically annotated knowledge base of fact-checked claims (exposed through established schemas such as *schema.org*), together with an API able to rank matching claim candidates for any given utterance, thereby offering a kind of semantic annotation/tagging of unstructured content that is novel and goes beyond the current entity and relation extraction.

Combining basic principles of the WWW, open standards and interfaces, we provide access to the underlying data and the claim linking API to be integrated into third-party applications. For demonstration purposes, we also provide a Web application as a basic showcase.

2 SYSTEM DESCRIPTION

2.1 Background

ClaimLinker relies on the ClaimsKG knowledge base,[1] an RDF knowledge graph of fact-checked claims that enables structured queries about their truth values, authors, dates, related entities and other contextual metadata [12]. ClaimsKG is generated through a pipeline which periodically harvests data from highly reputable fact-checking portals, including Politifact.com and Snopes.org. It currently contains 34,201 fact-checked claims published since 1996. The claims and their review articles are annotated with entities

[1] https://data.gesis.org/claimskg/

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8313-4/21/04.
https://doi.org/10.1145/3442442.3458601

from DBpedia and described by a specific RDFS model based on established vocabularies such as *schema.org* [9] and NIF [10]. Also, a normalised truth ratings scheme is introduced, containing four generic categories (*true*, *false*, *mixed*, *other*). Federated SPARQL queries and dedicated tools, such as the ClaimExplorer[2] or the ClaimsKG Statistical Observatory[3] enable exploration of the claim data and information discovery [7].

2.2 Problem Definition and Aim

Relating a piece of text to fact-checked claims, as those described in the ClaimsKG knowledge base, essentially comes down to the task of identifying links between utterances. In the context of claims, we isolate three main types of relations:

(a) *identity*, i.e. the two claim utterances are semantically equivalent (have the exact same meaning) and hence can be linked via an equivalence relation such as `owl:sameAs`;

(b) *similarity*, i.e. the two claims utterances share certain semantic proximity on a scale between 'identical' (represented by the same-as relation) and 'dissimilar'. This notion relates to that of semantic similarity discussed, for example, in [8] and tackled in the Semantic Textual Similarity task [1].

(c) *relatedness*, as opposed to similarity, covers "any kind of lexical or functional association" [8]. It thus encloses various relationships, such as meronymy, antonymy, logical or textual entailment [5].

The aim of the ClaimLinker application is to annotate a given text with fact-checked claims of ClaimsKG, with a focus on the type (b) relation described above (*semantic similarity*). More formally: given a text or document d and a knowledge base of fact-checked claims K, ClaimLinker aims at providing a set of claim annotations A of the form $\langle t, p, C \rangle$, where t is a piece of text in d (e.g., a sentence), p is the position of t in d, and C is a ranked list of semantically similar fact-checked claims of K. In particular, a claim $c \in C$ is of the form $\langle u, s \rangle$, where u is a claim URI from K and s is the similarity score representing the strength of the connection between c and t.

For example, consider the below text:

"Obama can complain about Republicans in the House as much as he wants. But in the first two years, he had, you know, huge majorities in the House and Senate, and did nothing with them to create jobs."

ClaimLinker links the second sentence (*"But in the first two years, he had, you know, ..."*) to the ClaimsKG URI <http://data.gesis.org/claimskg/creative_work/34465e1b-7108-53f7-8960-4595b8dd09b2> (Fig. 1). Using this URI, the requester (e.g., a user or a third-party application) can retrieve additional data from ClaimsKG, like related entities or other claims uttered by the same person.

2.3 Implementation Details

We consider a use case scenario where real-time claim linking is required. For example, a user reads an online news article and wants to find out if a statement that is mentioned in the article is related to one or more fact-checked claims. To this end, we consider a claim linking approach that comprises one (offline) pre-processing step (for indexing ClaimsKG) and four real-time processing steps.

Indexing and search service provision. We consider the data model of ClaimsKG [12] for indexing claim related data. In particular, we index the data using Elasticsearch[4] and setup a dedicated search service. The use of Elasticsearch offers fast candidate generation and scalability. We index the following fields of each claim: i) its URI, ii) its text, iii) its truth-value, iv) its author name, v) the URL and vi) the title (headline) of the corresponding fact-checking article. We use the text of the claim and the article headline as searching fields for candidate generation (more below), while the other fields are used for showing more information for each claim in the results.

Detection of check-worthy text. Here we need to find pieces of text in the input that can be linked to fact-checked claims. We consider all text sentences as check-worthy and filter out those for which the *candidate generation* step does not retrieve claims with adequate relevance score (more below). This allows us to avoid the production of false negative cases. Moreover, in one of our use case scenarios (Sect. 3) the user selects the text to be checked, thus we always need to consider the selected text as check worthy.

Candidate generation. Given a check-worthy sentence from the document, we submit a search query to the Elasticsearch service and get a ranked list of top-30 results. Each result here represents a fact-checked claim and is a candidate claim annotation for the input text. We use a *multi match* query that matches the input text to the indexed search fields of a claim (claim text, article headline) using the default similarity model of Elasticsearch (*Okapi BM25*). We filter out claims with an Elasticsearch relevance score less than 5, that has shown empirically to filter out very irrelevant claims.

Candidate ranking. We need a very fast ranking approach in order to support real-time annotation. As a baseline method, we consider a lightweight and fully unsupervised approach which considers both the relevance score returned by Elasticsearch and a set of eight simple and well-established textual similarity measures: common (based on Jaccard similarity coefficient between the input text and a candidate claim) *words*, *lemmatized words*, *named entities* (using Stanford CoreNLP [11]), *disambiguated entities* (using FEL [3]), *POS tags* (nouns, proper nouns, adverbs, verbs), *2/3/4-grams*, *2/3/4-chargrams*, and *cosine similarity*. For computing the final score of a candidate claim, we average the relevance score of the candidate claim as returned by Elasticsearch and the average of all the scores returned by the eight textual similarity measures.

2.4 Evaluation results

Experiments using the ground truth dataset of CLEF2020 CheckThat! 2020 Task 2 (*verified claim retrieval*) [2] demonstrate the effectiveness ClaimLinker in finding the correct fact-checked claim for a given text. In particular, the correct claim is returned in the first position (Precision@1) in 76% of the test cases, outperforming all unsupervised methods that participated in the challenge [2]. Also, the correct claim is returned in one of the top-3 positions in 80% of the test cases, and in one of the top-5 positions in 83.5% of the test cases. These are promising results given that we consider a fully unsupervised approach which does not require training or tuning and is very efficient. If we consider only the score returned

[2] https://data.gesis.org/claimskg/explorer
[3] https://data.gesis.org/claimskg/observatory
[4] https://www.elastic.co/

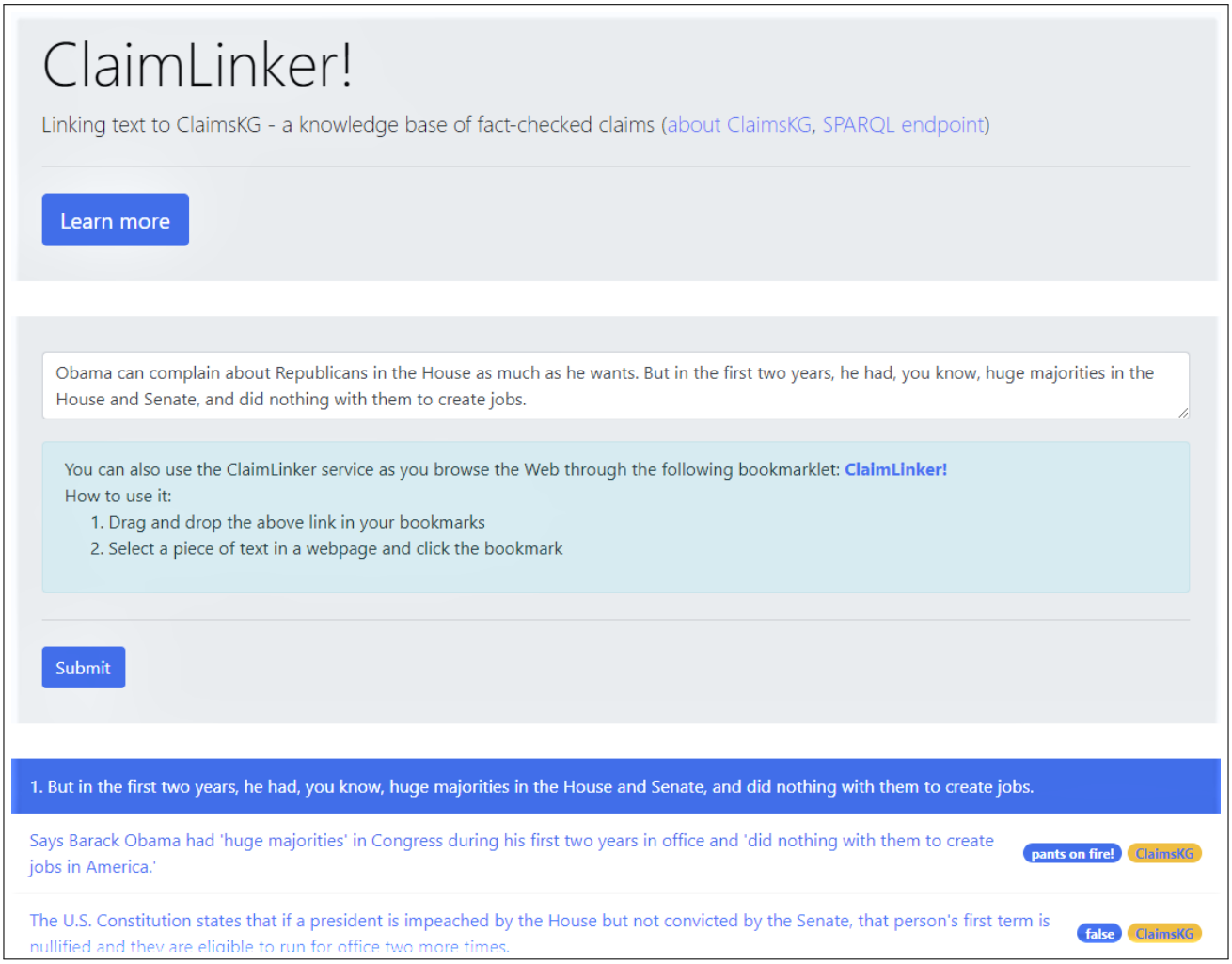

Figure 1: A screen shot of the ClaimLinker Web application.

by Elasticsearch, then performance (Precision@1) drops to 71.5%, while if we consider only the eight similarity measures performance drops to 62%. This showcases the benefit of combining the different scores. Finally, we can achieve higher precision by considering a higher Elasticsearch relevance score when filtering out irrelevant claims in the candidate generation step. For instance, by using a threshold score = 20 (instead of 5), Precision@1 is increased to 89.5%, however then recall decreases (the system does not provide any annotations for 28.5% of the test cases).

3 SYSTEM COMPONENTS AND USE CASES

We consider the following use cases (exploitation scenarios), supported by dedicated ClaimLinker services:

Web application[5] offering a form where the user can give some text and check if there are fact-checked claims linked to that text (Fig. 1). Use-case: a user visits the Claim Linker web page and checks if a statement is linked to fact-checked claims.

Bookmarklet[6] which allows a user to select a piece of text in any web page and check if there are fact-checked claims linked to the selected text. Use-case: a user reads a news article, selects a piece of text in the article and clicks the bookmarklet for checking for any linked fact-checked claim.

Java library[7] that communicates with an Elasticsearch service and offers the claim linking functionality. Use-case: the Java library is used by another Java project for linking text to fact-checked claims.

Web service[8] which accepts HTTP requests and returns the results in JSON. Use-case: the service is exploited by third-party applications for linking text to fact-checked claims. Through the claim URIs

[5] http://users.ics.forth.gr/~fafalios/claimlinker/

[6] Usage guidelines at: http://users.ics.forth.gr/~fafalios/claimlinker/

[7] https://github.com/malvag/ClaimLinker/tree/master/ClaimLinker_commons

[8] An example request and response is available at https://github.com/malvag/ClaimLinker#usage.

and the SPARQL endpoint of ClaimsKG, one can retrieve additional information about the linked claims.

The source code of all services is available on GitHub.[9]

4 CONCLUSION AND FUTURE WORK

We presented ClaimLinker, a Web service and API that links arbitrary text to a knowledge graph of fact-checked claims. Given a text snippet, for instance, a statement in a news item, ClaimLinker first obtains candidate claims from the knowledge graph through a preliminary blocking step and then ranks them using a range of similarity metrics, outperforming a number of unsupervised baselines on the claim retrieval task.

As applications of our work, we consider any use case where on-the-fly fact-checking is of importance and has to be streamlined by exploiting already fact-checked claims, for instance, as part of online journalism, social media or news platforms. Exploiting the semantic annotations of the underlying knowledge graph [12] facilitates further exploration, for instance, to identify claims related to the same or similar entities, or claims originating from the same sources.

Considering that our current approach utilises basic text similarity metrics for re-ranking candidates, future work will be concerned with more elaborate approaches, for instance, using self-supervised, transformer-based language models [6] to obtain contextual vector representations of both claim and text candidates in order to better cater for vocabulary mismatch problems. In addition, while we opted for an efficient unsupervised approach which determines weights of distinct metrics based on heuristics, a natural step is to attempt learning optimal weights, leading into supervised models for claim linking. Given that both approaches have advantages and disadvantages, thorough evaluation has to provide experimental insights into over-/underfitting problems and potential to generalise well across claims and types of corpora.

REFERENCES

[1] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* Association for Computational Linguistics, San Diego, California, 497–511.

[2] Alberto Barrón-Cedeño, Tamer Elsayed, et al. 2020. Overview of CheckThat! 2020 — Automatic Identification and Verification of Claims in Social Media. In *Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction.* Springer International Publishing, Cham, 215–236.

[3] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.* Association for Computing Machinery, New York, NY, USA, 179–188.

[4] Katarina Boland, Pavlos Fafalios, Andon Tchechmedjiev, Konstantin Todorov, and Stefan Dietze. 2019. Modeling and Contextualizing Claims.. In *Proceedings of the Blockchain enabled Semantic Web Workshop (BlockSW) and Contextualized Knowledge Graphs (CKG) Workshop co-located with the 18th International Semantic Web Conference, BlockSW/CKG@ISWC 2019, Auckland, New Zealand, October 27, 2019.* CEUR-WS.org.

[5] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop.* Springer, Berlin, Heidelberg, 177–190.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Malo Gasquet, Darlene Brechtel, Matthaus Zloch, Andon Tchechmedjiev, Katarina Boland, Pavlos Fafalios, Stefan Dietze, and Konstantin Todorov. 2019. Exploring Fact-checked Claims and their Descriptive Statistics. In *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019.* CEUR-WS.org.

[8] Jorge Gracia and Eduardo Mena. 2008. Web-based measure of semantic relatedness. In *International Conference on Web Information Systems Engineering.* Springer, Berlin, Heidelberg, 136–150.

[9] Ramanathan V Guha, Dan Brickley, and Steve Macbeth. 2016. Schema. org: evolution of structured data on the web. *Commun. ACM* 59, 2 (2016), 44–51.

[10] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. 2013. Integrating NLP using linked data. In *International semantic web conference.* Springer, Berlin, Heidelberg, 98–113.

[11] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations.* Association for Computational Linguistics, Baltimore, Maryland, 55–60.

[12] Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A Knowledge Graph of Fact-Checked Claims. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings.* Springer, Berlin, Heidelberg, 309–324.

[9] https://github.com/malvag/ClaimLinker