# SIRIUS-LTG: An Entity Linking Approach to Fact Extraction and Verification

**Farhad Nooralahzadeh and Lilja Øvrelid**
Department of Informatics
University of Oslo, Norway
{farhadno,liljao}@ifi.uio.no

## Abstract

This article presents the SIRIUS-LTG system for the Fact Extraction and VERification (FEVER) Shared Task. It consists of three components: 1) *Wikipedia Page Retrieval*: First we extract the entities in the claim, then we find potential Wikipedia URI candidates for each of the entities using a SPARQL query over DBpedia 2) *Sentence selection*: We investigate various techniques i.e. Smooth Inverse Frequency (SIF), Word Mover's Distance (WMD), Soft-Cosine Similarity, Cosine similarity with unigram Term Frequency Inverse Document Frequency (TF-IDF) to rank sentences by their similarity to the claim. 3) *Textual Entailment*: We compare three models for the task of claim classification. We apply a Decomposable Attention (DA) model (Parikh et al., 2016), a Decomposed Graph Entailment (DGE) model (Khot et al., 2018) and a Gradient-Boosted Decision Trees (TalosTree) model (Sean et al., 2017) for this task. The experiments show that the pipeline with simple Cosine Similarity using TFIDF in sentence selection along with DA model as labelling model achieves the best results on the development set (F1 evidence: 32.17, label accuracy: 59.61 and FEVER score: 0.3778). Furthermore, it obtains 30.19, 48.87 and 36.55 in terms of F1 evidence, label accuracy and FEVER score, respectively, on the test set. Our system ranks 15th among 23 participants in the shared task prior to any human-evaluation of the evidence.

## 1 Introduction

The Web contains vast amounts of data from many heterogeneous sources, and the harvesting of information from these sources can be extremely valuable for several domains and applications such as, for instance, business intelligence. The volume and variety of data on the Web are increasing at a very rapid pace, making their use and processing increasingly difficult. A large volume of information on the Web consists of unstructured text which contains facts about named entities (NE) such as people, places and organizations. At the same time, the recent evolution of publishing and connecting data over the Web dubbed "Linked Data" provides a machine-readable and enriched representation of many of the world's entities, together with their semantic characteristics. These structured data sources are a result of the creation of large knowledge bases (KB) by different communities, which are often interlinked, as is the case of DBpedia (Lehmann et al., 2015) [1], Yago [2] (Suchanek et al., 2007) and FreeBase [3] (Bollacker et al., 2008). This characteristic of the Web of data empowers both humans and computer agents to discover more concepts by easily navigating among the datasets, and can profitably be exploited in complex tasks such as information retrieval, question answering, knowledge extraction and reasoning.

Fact extraction from unstructured text is a task central to knowledge base construction. While this process is vital for many NLP applications, misinformation (false information) or disinformation (deliberately false information) from unreliable sources, can provide false output and mislead the readers. Such risks could be properly managed by applying NLP techniques aimed at solving the task of *fact verification*, i.e., to detect and discriminate misinformation and prevent its propagation. The Fact Extraction and VERification (FEVER) shared task [4] (Thorne et al., 2018) addresses both problems. In this work, we introduce a pipeline system for each phase of the FEVER shared task. In our pipeline, we first identify entities in a given claim, then we extract candidate Wikipedia pages for each of the entities and the most similar sen-

[1] dbpedia.org
[2] www.mpi-inf.mpg.de/yago/
[3] www.freebase.com
[4] www.fever.ai



*Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 119–123
Brussels, Belgium, November 1, 2018. 

tences are obtained using a textual similarity measure. Finally, we label the claim with regard to evidence sentences using a textual entailment technique.

## 2 System description

In this section, we describe our system which consists of three components which solve the three following tasks: Wikipedia page retrieval, sentence selection and textual entailment.

### 2.1 Wiki-page Retrieval

Each claim in the FEVER dataset contains a single piece of information about an entity that its original Wikipedia page describes. Therefore we first extract entities using the Stanford Named Entity Recognition (StanfordNER) (Finkel et al., 2005). We observe that StanfordNER is sometimes unable to extract entity names in the claim due to limited contextual information like in example 1 below:

**Example 1** *A View to a Kill is an action movie.*
NER: *[]*
Noun-Phrases: *[A View to a Kill, an action movie]*

To tackle this problem, we also extract noun phrases using the parse tree of Stanford CoreNLP (Manning et al., 2014) and the longest multi-word expression that contains words with the first letter in upper case. This enables us to provide a wide range of potential entities for the retrieval process. We then retrieve a set of Wikipedia page candidates for an entity in the claim using a SPARQL (Prud'hommeaux and Seaborne, 2008) query over DBpedia, i.e. the structured version of Wikipedia.

The SPARQL query aids the retrieval process by providing a list of candidates to the subsequent system components, particularly when the claim is about *film, song, music album, bands and etc*. Listing 1 shows the query employed for the entity *Meteora* in Example 2 below, which outputs the resulting Wikipedia pages (Pages):

**Example 2** *Meteora is not a rock album.*
Entity: *[Meteora]*
Pages: *['Meteora', 'Meteora (album)', 'Meteora Monastary', 'Meteora (Greek monasteries)', 'Meteora (film)']*

The query retrieves all the Wikipedia pages which contains the entity mention in their title.

```
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix fn:<http://www.w3.org/2005/xpath-functions/#>

SELECT DISTINCT ?resource
    WHERE {?resource rdfs:label ?s.
    ?s <bif:contains> '"Meteora"'.

    FILTER (lang(?s) ="en")
    FILTER ( fn:string-length
    (fn:substring-after(?resource,
    "http://dbpedia.org/resource/"))>1)
    FILTER (regex(str(?resource),
    "http://dbpedia.org/resource")
    && !regex(str(?resource),
    "http://dbpedia.org/resource/File:")
    && !regex(str(?resource),
    "http://dbpedia.org/resource/Category:")
    && !regex(str(?resource),
    "http://dbpedia.org/resource/Template:")
    && !regex(str(?resource),
    "http://dbpedia.org/resource/List")
    && !regex(str(?resource), "(disambiguation)")
        )
    }
```

Listing 1: SPARQL query to extract Wikipedia page candidates for entity mention (e.g. *Meteora*)

### 2.2 Sentence Selection

Given a set of Wikipedia page candidates, the similarity between the claim and the individual text lines on the page is obtained. We here experiment with several methods for computing this similarity:

**Cosine Similarity using TFIDF:** Sentences are ranked by unigram TF-IDF similarity to the claim. We modified the fever-baseline code to consider the candidate list from the Wiki-page retrieval components.

**Soft-Cosine Similarity:** Following the work of Charlet and Damnati (2017), we measure the similarity between the candidate sentences and the claim. This textual similarity measure relies on the introduction of a relation matrix in the classical cosine similarity between bag-of-words. The relation matrix is calculated using the *word2vec* representations of words.

**Word Mover's Distance(WMD):** The WMD distance "measures the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to 'travel' to reach the embedded words of another document" (Kusner et al., 2015). The *word2vec* embeddings is used to calculate semantic distances of words in the embedding space.

**Smooth Inverse Frequency (SIF):** We also create sentence embeddings using the SIF weighting scheme (Arora et al., 2017) for a claim and candi-

date sentences. Then we calculate the cosine similarity measure between these embedding vectors.

### 2.3 Entailment

The previous component provides the most similar sentences as an evidence set for each claim. In this component, the aim is to find out whether the selected sentences enable us to classify a claim as being either *SUPPORTED*, *REFUTED* or *NOT ENOUGH INFO*. In cases where multiple sentences are selected as evidence, their strings are concatenated prior to classification. If the set of selected sentences is empty for a specific claim, due to the failure in finding related Wiki-page, we simply assign *NOT ENOUGH INFO* as an entailment label. In order to solve the entailment task we experiment with the use of several existing textual entailment systems with somewhat different requirements and properties. We follow the instruction from the Git-Hub repositories of the three following models and investigate their performances in the FEVER textual entailment subtask:

**Decomposable Attention (DA) model (Parikh et al., 2016):** We used the publicly available DA model [5] which is trained on the FEVER shared task dataset. We asked the model to predict an inference label for each claim based on the evidence set which is provided by the *sentence selection* component.

**Decomposed Graph Entailment (DGE) model:** Khot et al. (2018) propose a decomposed graph entailment model that uses structure from the claim to calculate entailment probabilities for each node and edge in the graph structure and aggregates them for the final entailment computation. The original DGE model [6] uses Open IE (Khot et al., 2017) tuples as a graph representation for the claim. However, it is mentioned that the model can use any graph with labeled edges. Therefore, we provide a syntactic dependency parse tree using the Stanford dependency parser (Manning et al., 2014) which outputs the Enhanced Universal Dependencies representation (Schuster and Manning, 2016) as a graph representation for the claim.

**Gradient-Boosted Decision Trees model:** We also experiment with the *TalosTree* model [7] (Sean et al., 2017) which was the winning system in the Fake News Challenge (Pomerleau and Ra, 2017). The *TalosTree* model utilizes text-based features derived from the claim and evidences, which are then fed into Gradient Boosted Trees to predict the relation between the claim and the evidences. The features that are used in the prediction model are word count, TF-IDF, sentiment and a singular-value decomposition feature in combination with word2vec embeddings.

[5] https://github.com/sheffieldnlp/fever-baselines

[6] https://github.com/allenai/scitail

[7] https://github.com/Cisco-Talos/fnc-1

| | Evidence | | |
|---|---|---|---|
| **Similarity** | Precision | Recall | F1 |
| **Co**sine Similarity using TFIDF | **21.14** | **67.24** | **32.17** |
| **So**ft-Cosine Similarity | 19.50 | 65.53 | 30.05 |
| **Wo**rd Movers Distance (WMD) | 18.24 | 59.29 | 27.90 |
| **Sm**ooth Inverse Frequency (SIF) | 15.19 | 50.33 | 23.33 |
| **FE**VER Baseline | - | - | 17.18 |

Table 1: Evidence extraction results on development set

## 3 Experiments

### 3.1 Dataset

The shared-task (Thorne et al., 2018) provides an annotated dataset of 185,445 claims along with their evidence sets. The shared-task dataset is divided into 145,459 , 19,998 and 19,998 train, development and test instances, respectively. The claims are generated from information extracted from Wikipedia. The Wikipedia dump (version June 2017) was processed with Stanford CoreNLP, and the claims sampled from the introductory sections of approximately 50,000 popular pages.

### 3.2 Evaluation

In this section we evaluate our system in the two main subtasks of the shared task: I) *evidence extraction* (wiki-page retrieval and sentence selection) and II) *Entailment*. Since, the scoring formula in the shared-task considers only the first 5 predicted sentence evidences, we choose 5-most similar sentences in the sentence selection phase (Section 2.2).

#### 3.2.1 Evidence Extraction

Initially, the impact of different similarity measures in sentence selection is evaluated. Table 1 shows the results of the various similarity measures described in section 2 for the evidence extraction subtask on the development set. The re-

| Model | Label | F1 | | | Label Accuracy | FEVER Score |
|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | | |
| DA | NOT ENOUGH INFO | 46.00 | 10.00 | 17.00 | **50.61** | **37.78** |
| | REFUTES | 61.00 | 60.00 | 60.00 | | |
| | SUPPORTS | 46.00 | 82.00 | 59.00 | | |
| DGE | NOT ENOUGH INFO | 41.00 | 5.00 | 8.00 | 42.24 | 30.31 |
| | REFUTES | 62.00 | 30.00 | 41.00 | | |
| | SUPPORTS | 38.00 | 92.00 | 54.00 | | |
| TalosTree | NOT ENOUGH INFO | 28.00 | 1.00 | 3.00 | 44.93 | 31.54 |
| | REFUTES | 66.00 | 42.00 | 51.00 | | |
| | SUPPORTS | 40.00 | 92.00 | 55.00 | | |
| FEVER baseline* | | | | | 51.37 | 31.27 |

Table 2: Pipeline performance on the dev set with the sentence selection module. (*) In the FEVER baseline the label accuracy uses the annotated evidence instead of evidence from the evidence extraction module.

sults suggest that the simple cosine similarity using TF-IDF is the best performing method for the sentence selection component when compared to the other similarity techniques. With an F1-score of 32.17 it clearly outperforms the Soft-Cosine Similarity (F1 30.05), WMD (F1 27.90) and SIF (F1 23.22) measures. This component clearly also outperforms the FEVER baseline for this subtask (F1 17.18).

#### 3.2.2 Entailment

This component is trained on pairs of annotated claims and evidence sets from the FEVER shared-task training dataset. We here train two different models i.e. *DGE* and *TalosTree* and we utilize the pre-trained *DA* model. We evaluate classification accuracy on the development set, assuming that the evidence sentences are extracted in the evidence extraction phase with the best performing setup. The results are presented in Table 2 and show that the *NOT ENOUGH INFO* class is difficult to detect for all three models. Furthermore, the *DA* model achieves the best accuracy and FEVER score compared to the others. We also observe that the label accuracy has a significant impact on the total FEVER score.

#### 3.2.3 Final System

The final system pipeline is established with the *SPARQL query* and *cosine similarity using TFIDF* in the evidence extraction module, and using the decomposable attention model for the entailment subtask. Table 3 depicts the final submission results over the test set using our system.

| Similarity | Evidence | | | FEVER | |
|---|---|---|---|---|---|
| | P | R | F1 | Acc. | Score |
| Our System | 19.19 | 70.72 | 30.19 | 48.87 | 36.55 |
| FEVER Baseline | - | - | 18.26 | 48.84 | 27.45 |

Table 3: Final system pipeline results over test set.

## 4 Conclusion

We present our system for the FEVER shared task to extract evidence from Wikipedia and verify each claim w.r.t. the obtained evidence. We examine various configurations for each component of the system. The experiments demonstrate the effectiveness of the TF-IDF cosine similarity measure and decomposable attention on both the development and test datasets.

Our future work includes: 1) to implement a semi-supervised machine learning method for evidence extraction , and 2) to investigate different neural architectures for the verification task.

## References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Delphine Charlet and Geraldine Damnati. 2017. Simbow at semeval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 315–319. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 957–966. JMLR.org.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255. Association for Computational Linguistics.

Dean Pomerleau and Delip Ra. 2017. Fake news challenge. http://fakenewschallenge.org/,.

Eric Prud'hommeaux and Andy Seaborne. 2008. SPARQL Query Language for RDF. W3C Recommendation. http://www.w3.org/TR/rdf-sparql-query/.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

Baird Sean, Sibley Doug, and Pan Yuxi. 2017. Talos targets disinformation with fake news challenge victory. http://blog.talosintelligence.com/2017/06/talos-fake-news-challenge, Accessed: 2018-07-01.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA. ACM.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*.