Latent Signals for Identifying Fake News

Swati Arora
swati.arora@cs.ucla.edu

Pranav Sodhani
sodhanipranav@cs.ucla.edu

Shubham Mittal
mitshubh@gmail.com

Omkar Patil
omkar.9194@gmail.com

Heenal Doshi
heenal.doshi@gmail.com

ABSTRACT

The project focuses on identifying news articles as fake or real based on the content of the article. In the past, there has been a significant research done on building a fake news classifier using external signals such as credibility and trust rank. However, the literature is devoid of many ideas when an article needs to be identified as fake or not solely based on its content. Furthermore, recently, the issue of fake news article circulation has been making rounds with its effects being anticipated in the 2016 US presidential elections. Consequently, there has been a surge in the efforts for solving this problem. Given the plethora of such sources and the fickle nature of the source credibility, we decided to approach this problem my making use of linguistic patterns and identifying correctness of probable facts presented in the article. The central task was to identify potential features using which we can distinguish between real and fake articles. After brainstorming over all possible features, we decided to use features that exploit the semantic and syntactic information in an article and the knowledge base oriented characteristics of a given training news article.

It is apparent by this point that the goals of the project are ambitious and nature open ended. In spite of this, the accuracy we obtained using different approaches (knowledge base, topic modeling/semantic) was around 84% - 90%, while the accuracy obtained via other approaches (sentiment analysis) was too less to be relevant. Features were added incrementally to the model and discarded if they did not yield good results. In this project we have adopted a hybrid approach to perform the classification task. While the first approach focuses on performing semantic, syntactic and sentiment analysis of the article, the second approach focuses on verifying information which is presented as facts in an article using a knowledge graph.

Keywords

Fake news; linguistic patterns; syntactic, semantic and sentiment analysis; knowledge graph.

1. INTRODUCTION

Social media has, undoubtedly, become one of the most important news sources for a large audience. In the last 10 years, journalism has seen a cultural shift as more and more news and news readers have moved online [5]. According to [4], around 14% of Americans identified social media as their "most important" source of US election news. However, false stories easily penetrate and circulate amongst a large audience, thus forming misleading opinions. False or fake stories are stories which include incorrect facts and, unlike satirical stories, have been intentionally circulated to form public opinions. One such recent incident was observed during 2016 US elections where pro-Trump and pro-Clinton fake stories were generated and circulated to sway public opinions in their favor. A recent study [4] discussed this in detail, a database of fake stories circulated prior to the elections was created and a survey was conducted to estimate the share of Americans who read and believed the fake news. Their work estimates that the average voting age American saw, remembered and believed more than 0.71 pro-Trump fake stories and 0.18 pro-Clinton fake stories. Moreover, a meta-analysis covering more than 200 experiments shows that human subjects are only 4% better than chance in detecting lies in text [1]. The abundance of such stories also implies that human experts cannot be deployed to segregate true and fake stories. This has motivated research in the direction of designing algorithms to detect fake news articles.

A lot of approaches were suggested for tackling this problem, but a majority of them focused on exogenous features of the given input news articles, for instance the author, publisher, source etc. However, given the plethora of such sources and the fickle nature of the source credibility, we decided to solve the problem using the more robust, intuitive approach of using linguistic patterns, content of articles and similar approaches for solving the problem.

2. ANALYSIS AND PROBLEM FORMULATION

Given the constraint of using only textual content within an article, it is intuitive to use word/term distribution in an article for generating features. If we start talking about using word/term distribution or linguistic patterns for generating features, the most obvious approaches are those of using semantic analysis, sentiment analysis, syntactic analysis. Further, the other direction was to verify the correctness of facts presented in the article using a pre-built knowledge graph. We ended up approaching the problem in two different ways which are explained as follows:

2.1 Approach I: Building a Knowledge Graph

With the surge in fake news across the world, traditional methods of news verification by experts are unable to keep up. In this approach we build a knowledge graph which can be thought of as 'our view of the world'. Our approach was guided by the decision making process of the experts, who would classify a news as true or fake based on their prior knowledge. The data can be considered as a graph, consisting of objects/entities as nodes and the edges signifying the relationships between them. With this in mind, we model a gigantic graph representing the 'universal' truth. The nodes in the graph represent the entities (subjects/objects) in the data and the edges signify the relationship between the subject and the object. We follow the RDF standard [13] and represent facts in the form of (SPO) (subject, predicate, object) triplets, where subject and objects are entities and predicate represents the relation between the entities. For example, the fact 'Obama was born in Hawaii' can be represented using 2 nodes: Obama and Hawaii with a directional edge connecting them representing the predicate 'born in'. Different relation types can be represented by different predicates. We also obtain type hierarchies and type constraints from the RDF triplets. For example, Trump is the president, which in turn is a person and a living thing. A type constraint on the other hand tells us, that a person can only marry another person, not a non-living thing. We follow the LCWA (local closed world assumption) [10] when

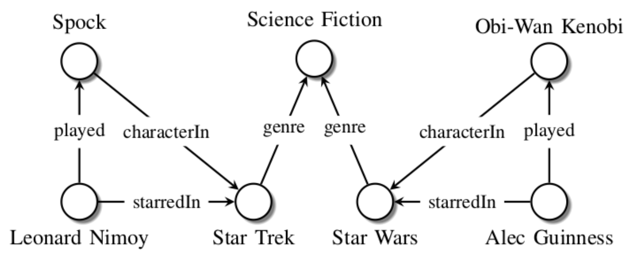

Figure 1: Small example knowledge graph where nodes represent entities, edges represent relationships and edge label indicate relation type

identifying the truthfulness of a new/missing triplet. The existing SPO triplets in the knowledge graph represent facts and for new triplets, and we could have labeled all triplets that do not belong in the graph to be false (corresponds to open-world assumption). This is a dangerous approach as our knowledge base is not entirely complete and has a lot of missing information. The LCWA is more accurate in the sense that it assumes the Knowledge Graph to be locally complete. In other words, if we have observed the triplet for a particular subject-predicate pair (s_i, p_i), then we assume that for any non-existing triplet of the form (s_i, p_i, .) is false. However, in the absence of a particular subject-predicate pair (s_i, p_i), we assume all triplets of the form (si, pi, .) to be unknown.

2.1.1 Methodology

We model the news verification problem as a link prediction task in the knowledge graph created above. Basically, finding a (s,p,o) triplet in the graph is the same as predicting the existence of an between subject s and object o. We alternatively mine paths between entities of the same type as the subject and object, and thus define the triplet in generalized form. To effective mine alternate paths, we use both the entity type and the predicate type to validate the veracity of a news triplet. For example, consider the true statement that “Sacramento is the capital of California”. We first find the entity types for the triplets. Sacramento is of entity-type: city, which California is of entity-type: state. Given this information, we compute the meta-paths for other triplets that match the types hierarchy. For this example, we get Atlanta → Georgia, Austin → Texas as positive pairs which are connected by the capital_of predicate. We also get negative pairs like Los Angeles → California, New York City → New York which do not conform our query. We can easily generate these positive and negative pairs if any one unit of the triplet is missing. Finally, we use logistic regression to train the model.

2.2 Approach II: Feature Extraction based on Semantic and Syntactic Analysis

The intuition behind this approach is to extract features for the classifier based on structure of data, emotion which it sends across and local dependencies and syntactic information which is an underlying feature of the language. We explored three different categories of features under this particular approach which are described as follows:

2.2.1 Using Semantic Features

It is easy to understand that real news articles will have a particular concentration of words and a frequent group of words across all the articles wherever the news topics are the same. For example: Wherever the topic “moon” was concerned, in real news articles we can expect following words to appear together: “moon, landing, real, Apollo, proof, authentic, NASA, yes, government etc”. Vice a versa in fake news articles for the same topic i.e. “moon” you can expect “moon, landing, fake, conspiracy, hoax, no, etc”. We plan to leverage this property of news articles while using semantic features.
When building a language model, one approach which is extensively used in data mining is n-gram model. But n-gram model does not capture long distance dependencies between words in an article. In order to capture such dependencies, probabilistic topic modeling approach was used. The assumption behind this approach is that in a well-defined corpus, articles are going to be from a range of topics where each article can be a mix of few topics with a focus on one or two topics. Hence the problem here could be formulated as extracting features based on latent topic information from the corpus and constructing a probabilistic model based on topic word density and topic proximity. Expanding this problem, a little bit, it can be stated that we not only want to know what the percentage of a topic words are in a given news article; but also what percentage of the most frequent words for a given topic are present in the given article to classify it.

2.2.2 Using Sentiment features

Intuition behind this is that given a dataset, we believe that fake news article tries to convey either very positive or very negative sentiment to the reader. For example, if the sentiment of an article can range from -1 to 1, we would like to think for a fake article it would be more likely be near -1 and 1 then distributed near 0. As opposed to this, the real news articles tend to be as unbiased as possible. Thus we plan to use this bias as a possible feature for identification of real/fake news articles. The problem for this approach can be formulated as calculating features which would represent the likelihood of positive and negative emotion and a label associated with the probabilities a news article being fake/real given the bias measure for that article.

2.2.3 Using Syntactic features

Given the real and fake news data set, it can be said that certain grammar/syntax patterns appear consistently in fake news articles when compared with real news articles and vice versa is also true. The distribution of such structures across various articles can be tracked and can be used as features for classifying. A simple example could be that in a fake news article, false facts having grammatical structure similar to statements/facts are likely to appear as fake article tend to be biased and don't use objectivity/doubt while writing statements. Hence the problem here can be formulated as calculating certain features which represent the likelihood of a news article being real/fake given occurrence of certain grammatical patterns. We deploy Charniak Parser to evaluate the grammatical features from the sentences in the articles.

3. RELATED WORK

Previous research mostly focused on classifying fake news using exogenous signals such as the credibility of the source, trust rank, etc. [2] identifies a set of features such as number of ads, valid links and plagiarism to train a model for credibility estimation and fake news classification. However, given the plethora of content generators, it is infeasible to mark news as fake based on only such evidences. Another approach is to identify the difference between linguistic patterns in the text of fake stories and that of genuine reporting. This is then exploited to make a classification model. In [1], a hybrid model is structured using linguistic cue approaches and network based approaches to make a feasible fake news detection system. However, most of academic work focuses on identifying fake news using source credibility as one of their strongest features. We, in this work, propose an algorithm to detect trust-worthiness of a news item based on solely the textual content (latent signals).

Most of the papers that we came across regarding this work were not full-fledged research projects in themselves but rather a theoretical description of the steps carried out by students for identifying fake/real news articles as a course project or for a competition. Thus the relevant works/papers were not extensive and did not provide any concrete/definite solution to the problem but rather provided a brief outline of various possible solution approaches to the problem. This scarcity in full-fledged work can be a result of how recent the problem is.

In [2], Chiu et. al. provided a brief idea of all the solution which they briefly touched upon. The major drawback to their project and thus their result was that, they weren't using "authentic" fake news articles per se, but using a trigram model for generating fake news articles from a large text corpus of real news articles. Hence the fake news articles had poor grammatical structure, and other attributed which the authors specifically targeted for identifying these articles. In the end, their project mostly revolved around classification of news articles displaying properties of news articles generated by the trigram model. Thus the results that they obtained for the various approaches were not representative at all and did not match with ours. Similarly, the conclusions that authors drew on the basis of these results were also of little relevance to us.
For instance, the primary features of their project were derived from statistical analysis. They used type token ratios, lexical entropy of the news articles for classifying articles belonging to a particular writing style (in this case the writing style of the trigram model). For authentic fake news articles, the authors would be varying and thus writing style will to be varied and of little use for classification as the same author could write both real as well as fake news articles.They achieved an accuracy of 86% using the statistical features as their primary features.Davis et. al. utilized news articles generated by trigram model as substitutes for fake news articles [5]. However, unlike the previous approach, their primary feature was generated from trigger networks, a concept quite similar to our own idea of knowledge graph.
The basic concept of their implementation was to produce weighted graphs having length between nodes between a given lower bound and upper bound, and the nodes being words. The edges basically represented how likely a given word was to appear in an article given another word has already appeared. The distance between the words could also be varied for better accuracy. This graph was used to calculate a feature which was essentially the average weight of all edges. Accuracy of 82% was achieved with their approach. This approach also tends to cater to the classification of articles generated by the trigram but is generic enough to use used for other data sets.

McIntire [1] in his blog discussed a genuine data set of "authentic" fake news articles by scraping real news from New York Times, WSJ, Bloomberg, NPR, and the Guardian that was published in 2015 or 2016 and fake news articles from Kaggle. He created and preprocessed his dataset of about 10558 articles (50% real, 50% fake). His approach consisted of using the individual words in the news articles for classifying the news articles as real or fake. Hence he went with Naive Bayes classifier and used words having a minimum frequency. He used a count vectoriser for word-count data structure and grid-search functionality from Scikit for deciding the optimal combination of various parameters from the word-count knowledge for getting a cross-validation accuracy of 91.6%.

We looked at Knowledge bases that are freely available, accessible and not covering special domains, for example, Geology, rather those that incorporate general knowledge. Two of the openly available knowledge bases we considered for building the knowledge graph were DBPedia and YAGO. DBPedia is a community project that extracts structured, multi-lingual information from Wikipedia and makes it freely available. The information is automatically extracted from Wikipedia such as info-box tables, categorization, external links etc. The English knowledge base contains around 400 million facts and 3.7 million things. The project also maintains a live knowledge base (DBPedia Live) which is updated whenever a page on Wikipedia changes. Figure 1 shows an example graph from DBPedia. YAGO (Yet another great ontology) is a semantic knowledge base derived from Wikipedia, WordNet and GeoNames. Currently, YAGO has knowledge of more than 10 million entities (like persons, organizations, cities, etc.) and contains more than 120 million facts about these entities. It attaches a temporal and spatial dimension to many of its facts and confidence value to its relations. Accuracy of YAGO has been manually evaluated and proved to be about 95%.While, DBPedia stores facts as single triples and does not store additional information about facts such as confidence value or temporal aspects, YAGO stores spatial-temporally enhanced facts. Data is available in RDF format from both DBPedia and YAGO. Both are also query-able using SPARQL endpoint. However, DBPedia has a much higher degree of connectivity with other LOD (Linking open data) datasets.

4. ALGORITHMIC DESCRIPTION

4.1 Semantic Features

The idea behind this approach is that in a well-defined corpus, articles are going to be from a range of topics where each article is represented as random mixtures over latent topics. Each topic is characterized by a distribution over words. One of the most popular techniques of topic modelling is LDA (Latent Dirichlet Allocation). We have used topic modelling methods and the textual data to build a distribution of words across multiple topics which were identified. This gives us a topic-word list. For instance, real article will have a consistent and a unique concentration of words for a particular topic, while the fake articles will have a different concentration of words for a particular topic. We have used mallet API as it uses LDA to extract topic related information and used it build feature vector[7].

After feeding the articles through mallet, we extract 20 most popular topics and corresponding word list for each topic. Using the output of mallet, we evaluate two score: "topic_word_density" (the ratio of most frequent words belonging to a particular topic in a given article to the total no of words in a given article) and "topic_proximity" (probability of most frequently used words associated with a fake/real article topic are present in the article being classified). Corresponding to each 20 topics, these two scores are evaluated for each article, yielding in 40 feature vectors.

4.1.1 Algorithm

1. Preprocess the dataset, and remove all stopwords, convert to small case
2. On the entire dataset, use mallet to find 50 topics, words belonging to each topic and select the 20 most frequent words in a given topic.
3. Calculate topic_proximity i.e. for each topic we calculate the fraction of the 20 most common words for that topic covered by the article
4. Calculate topic_word_density i.e. for each topic we calculate the fraction of words in the article that belong to that topic.
5. Corresponding to 20 topics, 40 features are generated. Feed 40 such features to train the svm[8] along with the class of the articles being real/fake. Pass 40 features of any article from test set to svm[8] to predict the class label.

4.2 Syntactic Analysis

The distribution of such grammar and syntax structures across various articles are being used as features for classifying. These features are derived from the parse structure of the sentence. It is hypothesized that real sentences and fake sentences tend to have different grammatical structures in general. An objective measure of the grammaticality of a sentence can be obtained by running it through a statistical parser. The log-likelihood score returned by the parser can be used to judge the grammaticality of a sentence and thus determine whether it is fake or real. The Charniak Parser[6] was used for assessing the grammaticality of the articles under test. The BLLIP parser is a python wrapper given to Charniak Parser[6]. It is a re-ranking parser with first stage parser and second stage re-ranker. Utilizing this re-ranking feature, we are generating multiple parse trees for a sentence and obtaining a score corresponding to the best possible structure for each sentence. BLLIP parser gives the functionality to easily set the number of such possible structures that we want. Out of all these possible structures, we get the most probable sentence structure for each sentence and get a log likelihood for the entire article as the numerical score for that sentence, namely 'parser_score'. This parser_score will be further normalized for every sentence in a given file and yielding a score called as "pgram".

4.2.1 Algorithm

1.Consider each sentence in an article from S1 to Sn.
2. Let length of each sentence be defined as L1 to Ln.
3. Let their log score from the Charniak Parser be P(S1)
4. For the entire calculate Pgram which is defined as
Pgram = Li.(P(Si)) / Li where i varies from 1 to n.
Use Pgram as a feature along with the class label of the training article for training the svm.
Pass the Pgram value for the test article for predicting class label.

5. DATASETS

We used the DBPedia dataset[12] which consisted of factual triplets extracted from the 'infoboxes' on Wikipedia. DBPedia is a community based project which uses the Resource Description Framework (RDF) for representing the extracted information. The resulting knowledge base is split into infobox-facts (SPO triplets) and 'owl' entity type mappings. The English version of the DBPedia describes 4.58 million things out of which 4.22 million things are classified in a consistent ontology which results into over 22 million entities. For our knowledge-base, we used a

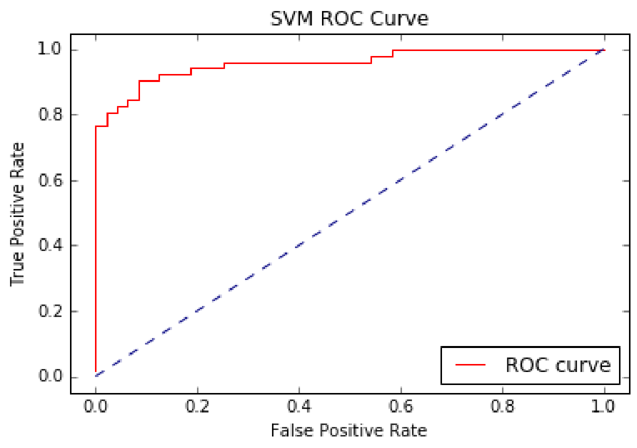

Figure 2: ROC curve based on Semantic and Syntactic features obtained on 1000-news dataset using SVM as a classifier

trimmed down version of the gigantic DBPedia dataset which included the cleaned infobox-facts and the ontology dataset from the DBPedia 2014 data dumps.

The dataset used for training and testing of the syntactic and semantic analysis methods was extracted from [14]. Here two data set primarily consists of two dat files, trainingset.dat and traininglabel.dat. The trainingset.dat file contains 1000 news articles with 500 articles belonging to each of real/fake category. The label of the articles is given in the traininglabel.dat . Both of these files are processed before being passed for feature generation from syntactic and semantic analysis. Processing of this data primarily consists of removing of stop-words, numbers, case conversion etc.

6. EXPERIMENT RESULTS

The final feature set used in our model includes 1 syntactic feature, 40 semantic feature and trigger network based on knowledge graph. Section 5.1 reports results obtained for classifying fake and real facts while using trigger networks based on knowledge graphs. Section 5.2 describes results on 41 feature set included incrementally by cross validation

6.1 Knowledge Base Approach

For the task of news verification, we transformed the problem into a link prediction problem as discussed before. test. The probability that an unknown statement is true is equivalent to the probability that the directed edge from subject to object is missing in the knowledge base. We perform the fact-verification task for a given predicate. To test the ability of our method to validate missing news-triplets and also model incoming new facts, we remove the given predicate and perform the verification task on the complement graph G' which is basically G – p. We perform 10-fold cross validation on the dataset. All the code in written in C++ and the test scripts from R. For the trimmed dataset we had over 4.6 million nodes with 100,001 relations of 671 different types. We borrowed code samples for optimization and test scripts from the implementation provided by authors of [11]. The test script contains the capital → state pairs. We define the relation between these pairs by the relation type 671, which corresponds to capital_of. For the test, we generate 200 incorrect random matchings of capitals to states. This leads to a total of 250 statements, out of which only 50 are correct. Using the above scripts, we obtain an accuracy of 84% with a 0.866 precision and a 0.840 recall value.

6.2 Semantic and Syntactic Approach

Features were added to the model in incremental approach. We first started with semantic features. Using mallet, we obtained 20 frequent topics and corresponding keywords. Using this information, topic_proximity and topic_word_density wascalculated corresponding to each topic, giving us a set of 40 features. Based on only these features, SVM was trained yielding an accuracy of 92%. Next set of features which were added were based on sentiment analysis corresponding to four features, probabilities for positive, negative and neutral sentiment and overall sentiment label for that article. We processed our dataset using many sentiment analyzers such as NLTK, Text Processing API. But the labels that we obtained for most of the articles were neutral yielding no bias toward any emotion. Thus it did not yield an important feature which could be integrated with our existing model. Next set of feature is obtained from syntactic analysis in which obtain parser_score which is a log-likelihood calculated over the entire article. On including syntactic, our model is trained on a complete set of 41 features. On combining all the 41 features, SVM model was run yielding an accuracy of 89%. Recall and Precision scores are illustrated in Table1. Area under curve(ROC) is shown in Figure 2.

Table 1: Classification report for 1000-news dataset using 10-fold cross validation and train-test split ratio of 9:1

Class	Precision	Recall	F1-score	Support
0 (Fake)	0.86	0.92	0.89	48
1 (Real)	0.92	0.87	0.89	52
Avg/Total	0.89	0.89	0.89	100

7. CONCLUSIONS

We believe knowledge graphs and bases are still in their nascent phase. We faced serious hardships in scaling the model and obtaining the database in an RDF format. Initially, we started by building our own graph from scratch. We parsed data from politifact.com in the form of *(s,p,o)* triplets, and implemented a similarity metric to cluster similar predicates. We used two 1-D hash maps to represent the subjects and the objects, and a 2-D hash set to represent the corresponding relations between subjects and objects. However, on deep introspection we realized we were unable to perform walks on the graph edges due to the inherent lack of type hierarchies in our dataset. Also, having a 2-D hash set severely dented our hopes of our scalable model. After an exhaustive literature review, we identified the vitality of a collaboratively curated dataset such as DBPedia. The immediate advantages that can be observed from using an RDF based format is that its heuristically inspired by the way, an article's veracity is judged by the experts. We generalize the problem using the type hierarchies provided by DBPedia and effectively model the context-dependency.
Thus we can see that with large enough data set, the text content of articles can be used to generate good enough features using semantic, syntactic analysis. Both of these analysis, although based on a few assumptions, yield surprisingly good accuracy. As far as syntactic analysis is concerned the pgrams score calculation took more time than expected. This process if parallelised, could be completed in much shorter time. Also these features were limited to data downloaded from the cmu coursewebsite. We did come across other data sources like Politifact but the used dataset was goodenough and just the right size to getthe requiredresults without consuming too much time. We firmly believe that if these features were combined with other features like that of Authors, Publisher, Sources even better accuracy can be obtained.

8. REFERENCES

[1] George McIntire (2016). https://opendatascience.com/blog/how-to-build-a-fake-news-classification-model/

[2] Chiu, J., Gokcen, A., Wang, W., & Yan, X. (2013). Classification of Fake and Real Articles Based on Support Vector Machines.

[3] Badaskar, S., Agarwal, S., & Arora, S. (2008). Identifying Real or Fake Articles: Towards better Language Modeling. In IJCNLP (pp. 817-822).

[4] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit.

[5] Davis, E., Adams, J., & Cohen, S. (2007). Classifying Articles as Fake or Real. Language and Statistics course project.

[6] Charniak, E. (2001, July). Immediate-head parsing for language models. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (pp. 124-131). Association for Computational Linguistics.

[7] Mallet API: http://mallet.cs.umass.edu/

[8] Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

[9] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago:A Core of Semantic Knowledge," in Proceedings of the 16th International Conference on World Wide Web.New York, NY, USA: ACM, 2007, pp. 697–706

[10] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "AMIE: Association Rule Mining Under Incomplete Evidence in Ontological Knowledge Bases," in Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 413–422.

[11] Shi, Baoxu, and Tim Weninger. "Discriminative predicate path mining for fact checking in knowledge graphs." *Knowledge-Based Systems* 104 (2016): 123-133.

[12] J. Lehmann, R. Isele, M. Jakob, DBPedia - a large-scale, multilingual knowledge base extracted from Wikipedia, Semantic Web 5 (1) (2014) 167–195.

[13] DL, OWL. "Resource description framework (RDF)." (2007).

[14] 1000-news dataset: http://www.cs.cmu.edu/~roni/11761-f16/project.html