Fact Checking from Natural Text with Probabilistic Soft Logic

Nouf Bindris, Saatviga Sudhahar and Nello Cristianini

Department of Computer Science, University of Bristol
nouf.bindris@bristol.ac.uk, saatviga.sudhahar@bristol.ac.uk,
nello.cristianini@bristol.ac.uk

Abstract. We demonstrate a method to support fact-checking of statements found in natural text such as online news, encyclopedias or academic repositories, by detecting if they violate knowledge that is implicitly present in a reference corpus. The method combines the use of information extraction techniques with probabilistic reasoning, allowing for inferences to be performed starting from natural text. We present two case studies, one in the domain of verifying claims about family relations, the other about political relations. This allows us to contrast the case where ground truth is available about the relations and the rules that can be applied to them (families) with the case where neither relations nor rules are clear cut (politics).

Keywords: Fact checking · Information Extraction · Probabilistic Soft Logic

1 Introduction

The vast availability of information on the web, its incompleteness, inconsistencies and the speed with which it spreads, have recently brought the need for identifying fake information. Detecting if an assertion is true or false is a tall order for an algorithm, as it may also be for a person, except for special cases where the assertion directly contradicts a known fact. Yet we expect algorithms to help us weed out fake news stories from online media [7, 18]. Fact checking, once the domain of journalists and editors, and now the realm of specialists, remains a time consuming and specialised task. We are interested in the situation where assertions must be assessed by an algorithm, without requiring an authoritative source of truth (a controversial requirement in the case of the press).

We will focus on assertions that we consider "implausible" because they implicitly conflict with a number of other statements present in a corpus of reference. For example, if all newspapers report various statements placing Hillary Clinton in the "pro-choice" camp of a debate, a single news item placing her in the "pro-life" camp would require further fact checking, and be deemed implausible, but not necessarily false. This approach allows us to handle statements that contain some degree of judgement, and not just expressions of facts, because we focus on the compatibility or internal consistency of large numbers of claims. In this paper we take the view that human fact checkers can benefit from a method which flags statements that do not naturally fit with a knowledge base, a corpus, or a set of rules, and are therefore implausible, or surprising. This could provide at least some degree of protection in the news ecosystem. Note that similar tools can be useful in many other scenarios, besides screening news in social media, for example they can be used to help curate large projects like Wikipedia, identifying claims in one page that conflict with claims in other pages.

The technical question of this paper is: how can we use techniques from information extraction and probabilistic reasoning to check facts that are implicit in a set of documents written in natural language? How can we decide if a claim is compatible with other claims, i.e. can it be true when the others are also true? Another way to formulate this question is: can we extract information that is not explicitly stated, but is implicitly present, in a set of documents?

We use two case studies to demonstrate the approach: one based on statements of fact and the other based on judgements. In the first case, we rely on natural language descriptions of the British Royal Family, and on facts about family relations, to extract the actual relations between members of the family, and use them to fact-check claims about further family relations (e.g. Who is whose cousin?). In the second case, we rely on news accounts of the 2012 US Elections, and general assumptions about how political relations work, to extract or check information about the political position of certain actors (e.g. Who supports which issue in the debate?). Technically, we make use of GATE [4] for information extraction, and Probabilistic Soft Logic [9] for inference. The documents are parsed, the named entities and their relations are extracted from natural language, then they are provided to the reasoning module that uses a knowledge base to see if a given claim is compatible with the rest.

In Section 2 we discuss related work in the domain of fact checking. In Section 3 we present Probabilistic Soft Logic (PSL) as a method of inference. In Section 4 we demonstrate our approach in the case of checking Family Relations. In Section 5 we demonstrate the approach in the case of checking Political Relations and in Section 6 we discuss limitations and future work.

2 Related work

Several automated fact-checking systems [1, 6, 7, 18] have been developed and used in real-world scenarios, such as monitoring false claims during the primary and general election debates throughout the 2016 U.S. elections. Given a claim, it is checked by first collecting supporting or opposing evidence from knowledge bases and the web, generate questions/queries related to the claim and a final answer derived and presented to the user based on discrepancies between the returned answers and the claim.

Fact checking numerical claims has also been studied in recent times. For example, Vlachos and Riedel [16] focused on fact checking simple numerical claims such as "population of Germany in 2015 was 80 million". They used distant supervision for identification and verification of claims to fact check 16 numerical properties of countries (such as population etc.). Input claims were matched with entries in a knowledge base and verdicts were deduced. In the follow-up work, they extended the system to include temporal expressions, so that the temporal context of the claim could be taken into account [15].

Recent work has used Markov Logic Networks to reason about the world under uncertainty, answering questions such as "According to sources A and B, is Mr. Doe euro-sceptic?" [10, 11]. Their algorithms support the task of extracting information about the facts from various sources and fact checking the claims against background data although it was not tested on real-world data. Work by Patwari et al. [12] discusses a system to identify check-worthy statements in political debates which needs to be fact-checked using a multi-classifier system that models latent groupings in data. These statements may not be explicitly mentioned in the text but they are check-worthy. From the statement, "We need the private sectors help, because government is not innovating" they identify a check-worthy claim such as "the U.S. government is not innovating". Natural language summaries of relational databases have also been fact-checked in a semi-automatic way using probabilistic modelling that identifies erroneous claims in articles from major newspapers [8]. The limitation in their work is that it requires humans to check the interpretations of the system and correct it if it was wrong.

In contrast, we check claims that are not explicitly stated in the text corpus. Using a knowledge base of extracted facts from various sources and first order logic rules we infer information that is implicit in text. We focus on detecting claims that can be considered as not plausible, in that they implicitly contradict background knowledge, assumptions or other claims contained in a reference corpus.

3 Probabilistic Soft Logic

Probabilistic soft logic (PSL) [9] is a framework that allows users to specify rich probabilistic models over continuous-valued random variables using first-order logic to describe features that define a Markov network similar to statistical relational learning languages such as Markov Logic Networks (MLNs). User-defined predicates model relationships and attributes and first-order logic rules model dependencies or constraints on these predicates in a PSL program. A PSL program consists of a set of predicates, weighted rules involving these predicates, and known truth values of ground atoms derived from observed data. Inference for the PSL program is over the remaining unknown truth values. PSL uses the most probable explanation (MPE) inference which is to find the most probable interpretation given evidence, that is, the most likely interpretation extending a given partial interpretation [9]. Given a set of atoms $l = \{l_1, ...l_n\}$, we call the mapping $I : l \to \{0, 1\}^n$ from atoms to *soft truth* values an *interpretation*.

Soft logic is mathematically represented in PSL using the Lukasiewicz t-norm as the relaxation of the logical AND and OR, respectively. These relaxations are exact at points, when variables are either true(1.0) or false (0.0), and provide a consistent interpretation for values in-between. The probability distribution defined by a PSL program measures the overall distance to satisfaction, which is a function of all ground rules truth values.

A PSL program containing a set of rules and ground atoms induces a distribution over interpretations I given by,

$$f(I) = \frac{1}{Z} exp[- \sum_{r \in R} \lambda_r (d_r(I))^p] \tag{1}$$

where λ_r is the weight of the rule r, Z is a normalization constant and $p \epsilon \{1, 2\}$ provides a choice of two different loss functions. $p = 1$ refers to satisfying one rule while $p = 2$ refers to satisfying all rules to some extent. These probabilistic models are said to be instances of Hinge-loss Markov random fields [2]. In our work, we use PSL because it's proven to be scalable and it works with continuous truth values which is useful for different modelling problems.

4 Fact checking Family Relations

In this study we use a long BBC news article describing kinship of the members in the royal family[1]. This includes a Royal Family tree and line of succession beginning from Queen Elizabeth II to Prince George. We automatically extract information from this article about family relations from the Royal Family such as Parent and Spouse. For example we extract,

Charles is the **Parent** of William
William is the **Spouse** of Kate

We build a knowledge base with the facts extracted and use logical rules in PSL to infer relationships not mentioned in text. How we extract facts is explained in Section 4.1. We then fact check claims about the Royal Family. PSL is a system for collective inference and therefore it can collectively infer new relationships according to logical rules specified. Eventually, we can check our claims against the system. If the result for the claim was already inferred by PSL the system returns the verdict, a binary value 0 (False) or 1 (True). If not the fact from the claim is added to PSL targets and the result is inferred. In the following sections we explain how we automatically extract facts from text, infer new relations not mentioned in text and then fact check similar claims.

4.1 Fact Extraction

We use ANNIE, a Nearly-New IE system in GATE [4], an open source platform for text engineering in order to extract named entities with their gender from text. We chose to use GATE since its simple, scalable and easily customisable with the use of JAPE grammars and Gazetteer lists. We do co-reference resolution, which is the process of determining whether two expressions in natural

[1] Royal Family tree and line of succession:
http://www.bbc.co.uk/news/uk-23272491

language refer to the same entity in the world [13]. For example, Queen Elizabeth II and Queen refer to the same entity. The Orthomatcher module in the ANNIE Information extraction system in GATE [13] is used to perform this task. We resolve pronouns to their referring entity names using the Pronominal resolution module. The system resolves pronouns such as 'he', 'she', 'his', 'him' and 'her' to their referring entity names. JAPE grammars are used to extract patterns of Parent and Spouse relations. For example, the grammar shown below says if a Person entity is followed by the word 'child' or 'son' or 'daughter' which is then followed by the word 'of' followed by a Person entity, the first person refers to a Parent entity. Therefore, the system annotates the relation as Parent relation. {Tokens} refer to pronouns and stop words that could occur inbetween.

$$\text{Person,\{Tokens\},Token} == (\text{“child”} \mid \text{“}son\text{”} \mid \text{“}daughter\text{”}),\\ \text{Token} == \text{“of”,Person}$$

Similarly we annotate Spouse relations if a Person entity is followed by the word 'married' or 'wife' or 'husband' which is then followed by another Person entity.

$$\text{Person,\{Tokens\},Token} == (\text{“married”} \mid \text{“}wife\text{”} \mid \text{“}husband\text{”}),\\ \text{\{Tokens\},Person}$$

We extracted 16 female names, 12 male names, 10 Parent relations and 7 Spouse relations from the article and this information was added to our knowledge base in PSL. In the next step we use logical rules to infer relations not explicitly mentioned in text.

4.2 Inferring Relations

From the extracted family relations, we infer relations that were not explicitly mentioned in text such as Cousins, Sisters, Brothers, Siblings, Uncle, Aunt, Niece and Nephew. Examples of a few logical rules we used to infer relations Cousins, Siblings, Uncle, Aunt and Nephew are shown below.

$$Parent(X,B) \wedge Parent(X,A) \wedge (A \neg = B) \Rightarrow Siblings(A,B)$$
$$Parent(X,B) \wedge Parent(Y,A) \wedge Siblings(X,Y) \Rightarrow Cousins(A,B)$$
$$Parent(X,B) \wedge Siblings(X,Y) \wedge Female(Y) \Rightarrow Aunt(Y,B)$$
$$Parent(X,B) \wedge Siblings(X,Y) \wedge Male(Y) \Rightarrow Uncle(Y,B)$$
$$Parent(X,B) \wedge Siblings(X,Y) \wedge Male(B) \Rightarrow Nephew(B,Y)$$

The first rule infers Siblings, saying that if X is the Parent of B and X is also the Parent of A and A and B are different people then B and A should be Siblings. The second rule says A and B are Cousins if X is the Parent of B, and Y is the Parent of A, X and Y are siblings. The third rule says if X is the Parent of B and X is the sibling of Y and Y is a Female then Y is the Aunt of B. The fourth rule infers Uncle relation and fifth Nephew relation.

PSL uses MPE inference to infer information, which is to find the most probable interpretation given evidence but also provides a lazy implementation of the algorithm. We use the Lazy MPE inference in PSL which allows to specify only the required targets for inference and uses less memory.

4.3 Fact Checking

In total the system inferred the following number of relations from text: 10 Cousins, 7 Uncles, 3 Aunts, 11 Siblings, 4 Nephews and 6 Nieces. We checked if the inferred relations were correct by manually checking the family tree given in the article, and all of them were correct.

When a new fact needs to be checked about family relations, it is checked against relations that are inferred already by PSL. If it was already inferred, the Verdict True or False is returned. Otherwise the fact is added to the target list in PSL, which then initiates the inference process and returns a result. Following examples show how a claim regarding Cousins and Nephew relation is converted to a target, added to PSL and how the Verdict True or False is returned.

Claim: "Is Prince William the Cousin of Princess Euginie"
Target : Cousin(Prince William, Princess Euginie)
Verdict: 1.0 / True

Claim: "Is Prince William the Nephew of Princess Beatrice"
Target : Nephew(Prince William , Princess Beatrice)
Verdict : = 0.0 / False

5 Fact checking Political Relations

In this study we infer and fact check political relations among actors in a political network generated from 130,213 English news articles about 2012 US Elections. This involves fact checking supporting or opposing views of Political actors towards other actors and issues. Data collection was done via extraction of news articles using a modular media content analysis system [5] containing US and International media and training a topic classifier to classify election articles.

5.1 Fact Extraction

We extract subject-verb-object (SVO) triplets from the election news collection via a fully automated pipeline [14] that performs named entity detection, coreference and anaphora resolution before the triplet extraction. In the triplets, subjects and objects are named entities or noun phrases (issues) and the verb expresses a positive or negative attitude between the subjects and objects in the political discourse. The number of triplets are reduced in size after filtering high confidence triplets and they are used to create positively and negatively weighted relations between actors. We make use of positive and negative verb lists to count a triplets as a vote in favour of a positive or negative attitude and calculate a weight for the relation between actors. Verb lists denoting political support/opposition were manually created by going through actions in triplets that were extracted from the elections corpus and labelling then positive or negative. When quantifying the weight of a relation between actors a

and b a confidence interval [17] around the estimate of the value is also considered. Based on computed confidence intervals, we extract relations that are sufficiently supported by the corpus, calculate positive and negative weights and use them to assemble a network consisting nodes representing actors/issues and edges representing the weights ranging from [-1 +1]. From this network we use structural balance [3] rules to infer political relations among actors and between actors and issues using PSL.

Structural balance can at most give us plausibility of a claim, as it is not an exact relation like family relations. An inferred political relation will have a weight corresponding to the level of support or opposition between actors in the relation conveying how plausible it is.

5.2 Inferring Relations

In order to prove that we could infer political relations among actors from the network, we remove a few links from it and use the remaining relations to predict the removed links. We want to see when 5%, 10% or even 20% of the links are removed from the network can we still infer them using the remaining observed relations. Since we have the truth values for the removed links we also evaluate the performance of the system. The network used for this study contains 169 nodes and 238 links with weights in the interval [-1,1]. To make this appropriate for the PSL framework, weights were normalized to [0,1] interval. First, we carefully select the number of links that should be removed from the network. This involves the links that connect nodes with a degree greater than or equal to 2 so that we do not introduce singletons in the network when links are removed. In total we quantify 126 links as removable.

We then remove 5% (12 links), 10% (24 links) and 20% (48 links) of the links from the whole network randomly selected from the 126 removable links identified and predict them using PSL. The logical rules created for predicting links are based on the structural balance theory [3] with a binary predicate *Rel* (relations between actors). A few logical rules are shown below.

$$\begin{gathered}
Rel(A,B) \wedge Rel(B,C) \Rightarrow Rel(A,C) \\
Rel(A,B) \wedge \neg Rel(B,C) \Rightarrow \neg Rel(A,C) \\
\neg Rel(A,B) \wedge Rel(B,C) \Rightarrow \neg Rel(A,C) \\
\neg Rel(A,B) \wedge \neg Rel(B,C) \Rightarrow Rel(A,C)
\end{gathered}$$

The first four rules adapt to structural balance in transitive triads in the network that state a friend of my friend is my friend, a friend of my enemy is my enemy, an enemy of my friend is my enemy and the enemy of my enemy is my friend. Political relations are not always transitive and therefore in future we plan to add more rules that can better explain the relationships between political entities. The outcome is a set of truth values assigned by PSL for links predicted and this relies on the input relations that are highly confident. Since we also know the truth values for the links predicted, in each case we measure the Mean Absolute Error (MAE) over all the links predicted in 100 iterations. The MAE over all predictions is given by,

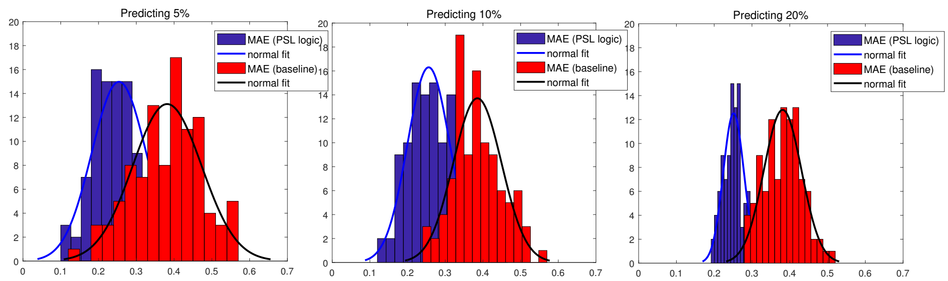

Fig. 1. MAE distribution with normal fitted curve over 100 iterations for the predictions with PSL and random (baseline) when predicting 5% (left), 10% (middle) and 20% (right) of the links from the network

$$MAE = \frac{\mid y^i - x^i \mid}{n} \quad (2)$$

where y^i refers to the prediction of the i^{th} link, x^i refers to the truth value of the i^{th} link and n, the total number of links predicted.

We compute the MAE over 100 iterations when removing 5%, 10% and 20% of the links from the network and predicting them with PSL. To compare this with a baseline and prove its better than random, in each experiment we randomly pick a value from the whole link weight distribution of the network as the prediction and compute the MAE as before. Figure 1 shows the MAE distribution with a normal fitted curve over 100 iterations for the predictions with PSL and random predictions (baseline) when predicting 5%, 10% and 20% of the links from the network. The most common MAEs lie in the range 0.19-0.27 (5%), 0.22-0.28 (10%), 0.25-0.28 (20%) for PSL predictions and 0.33-0.41 (5%), 0.33-0.39 (10%) and 0.34-0.39 (20%) for the baseline. Therefore the test does show that PSL does better than random in predicting relations.

5.3 Fact Checking

Now since we have proven that political relations could be inferred given a set of political relations between actors, we can use this to check facts about political relations.

For example given a claim/fact such as,

Claim: "Hillary Clinton opposes Abortion".

the system adds this fact to the PSL target list and runs the inference process to fact check the truth. The weight of this relation could be assigned to 0 since oppose is a negative verb in the context of elections and the most negative weights are mapped to 0 values in PSL. The target is comprised of Hillary Clinton, Abortion and the negative weight associated with the relation.

Target : (Hillary Clinton, Abortion)
Claim Weight: 0.0
Inferred Weight: 0.85
Verdict : = 0.0 / False

The inferred weight for the given target is 0.85 indicating that there is a reasonably high support for Abortion from Hillary Clinton. Comparing to the weight of Claim (0.0) the system returns the Verdict False. It is also possible to reason out this decision saying that Hillary Clinton supports Obama and Obama supports Abortion, therefore Clinton supports Abortion violating the first logical rule given to PSL which says if A supports B and B supports C, then A supports C.

6 Conclusion and Future work

This paper has demonstrated an automated system to detect claims that can be considered as not plausible, in that they implicitly contradict background knowledge, assumptions or other claims contained in a reference corpus. The key is that the claim we are checking is not explicitly stated in the reference corpus, and the necessary knowledge to verify it is potentially distributed across many documents. We address this by combining information extraction with probabilistic reasoning, to see if a claim can follow from other known facts showing two examples, fact checking Family relations for which ground truth is available and Political relations where neither relations nor rules are clearly available. We check the implausibility of claims in that domain. We expect this kind of approach to be useful for projects like Wikipedia, or to provide support to news fact checkers, but always in the form of assisting the job of humans. We are planning to deploy these tools to very large corpora combining information from multiple sources such as those created by digital humanities and computational social sciences as well as to applications that can lead to Q/A systems based on news content. The main challenge lies in scaling up the probabilistic reasoning to work with large amounts of facts while also having the ability to provide explanations to the verdicts given by the system.

Acknowledgements. NC and SS were supported by ERC, NB was supported by a grant from KSU, Saudi Arabia.

References

1. Ba, M.L., Berti-Equille, L., Shah, K., Hammady, H.M.: Vera: A platform for veracity estimation over web data. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 159–162. International World Wide Web Conferences Steering Committee (2016)
2. Bach, S., Huang, B., London, B., Getoor, L.: Hinge-loss markov random fields: Convex inference for structured prediction. arXiv preprint arXiv:1309.6813 (2013)

3. Cartwright, D., Harary, F.: Structural balance: a generalization of heider's theory. Psychological review **63**(5), 277 (1956)
4. Cunningham, H., Wilks, Y., Gaizauskas, R.J.: Gate: a general architecture for text engineering. In: Proceedings of the 16th conference on Computational linguistics-Volume 2. pp. 1057–1060. Association for Computational Linguistics (1996)
5. Flaounas, I., Lansdall-Welfare, T., Antonakaki, P., Cristianini, N.: The anatomy of a modular system for media content analysis. arXiv preprint arXiv:1402.6208 (2014)
6. Hassan, N., Adair, B., Hamilton, J.T., Li, C., Tremayne, M., Yang, J., Yu, C.: The quest to automate fact-checking. world (2015)
7. Hassan, N., Zhang, G., Arslan, F., Caraballo, J., Jimenez, D., Gawsane, S., Hasan, S., Joseph, M., Kulkarni, A., Nayak, A.K., et al.: Claimbuster: the first-ever end-to-end fact-checking system. Proceedings of the VLDB Endowment **10**(12), 1945–1948 (2017)
8. Jo, S., Trummer, I., Yu, W., Liu, D., Mehta, N.: The factchecker: Verifying text summaries of relational data sets. arXiv preprint arXiv:1804.07686 (2018)
9. Kimmig, A., Bach, S., Broecheler, M., Huang, B., Getoor, L.: A short introduction to probabilistic soft logic. In: Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications. pp. 1–4 (2012)
10. Leblay, J.: A declarative approach to data-driven fact checking. In: AAAI. pp. 147–153 (2017)
11. Leblay, J., Chen, W., Lynden, S.: Exploring the veracity of online claims with backdrop. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 2491–2494. ACM (2017)
12. Patwari, A., Goldwasser, D., Bagchi, S.: Tathya: A multi-classifier system for detecting check-worthy statements in political debates. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 2259–2262. ACM (2017)
13. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Computational linguistics **27**(4), 521–544 (2001)
14. Sudhahar, S., De Fazio, G., Franzosi, R., Cristianini, N.: Network analysis of narrative content in large corpora. Natural Language Engineering **21**(1), 81–112 (2015)
15. Thorne, J., Vlachos, A.: An extensible framework for verification of numerical claims. In: Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics. pp. 37–40. Association for Computational Linguistics (2017)
16. Vlachos, A., Riedel, S.: Identification and verification of simple claims about statistical properties. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2596–2601. Association for Computational Linguistics (2015)
17. Wilson, E.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association **22**(158), 209–212 (1927)
18. Wu, Y., Walenz, B., Li, P., Shim, A., Sonmez, E., Agarwal, P.K., Li, C., Yang, J., Yu, C.: icheck: computationally combating lies, d–ned lies, and statistics. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 1063–1066. ACM (2014)