Spotting Knowledge Base Facts in Web Texts

Tomasz Tylenda, Sarath Kumar Kondreddi, Gerhard Weikum
Max Planck Institute for Informatics, Saarbrücken, Germany
{ttylenda,skondred,weikum}@mpi-inf.mpg.de

Abstract

Knowledge bases (KB) such as DBpedia, YAGO and Freebase have been constructed by harvesting facts from high-quality data sources and incorporating community contributions. Accurately detecting occurrences of these KB facts in *complementary sources* (sources other than where they were extracted from) is crucial for fact validity assessments and deriving occurrence statistics. In this paper we consider fact spotting – the task of automatically discovering the mentions of KB facts in text documents. Our fact spotting methodology follows a two-stage approach. First, we perform similarity-based matching of noun phrases with labels of KB entities and dependency path structures with patterns of KB relations. Next, we perform joint matching of mentions, entities, paths, relations, and their textual locations by encoding them into variables of an integer linear program. We evaluate our method by spotting Freebase facts in biographies on the Web.

1 Introduction

Motivation. Current knowledge bases (KB) such as DBpedia [1], Freebase [3], and YAGO [9] contain relational facts on millions of entities. The construction of these large KBs is driven by employing automatic information extraction methods (IE) on a variety of sources, both structured (e.g., Wikipedia infoboxes, gazetteers) and unstructured (e.g., news sources, product descriptions). However, facts extracted through IE are not completely error-free; errors may result either from incorrect statements at the source or be induced by the extraction process. For these reasons, KBs typically retain the extraction context from the source as provenance information, which provides evidence for corrections by human curators.

In order to make validity assessments on IE extractions or user generated facts, human contributors not only rely on provenance information but also look for *complementary sources* that mention these facts. Such external sources could be news articles, biographies, sports columns, or product websites. This manual curation process can be greatly simplified by automatically analyzing the text within a document and reporting the presence or absence of facts. We refer to this task as *fact spotting*.

Fact spotting has several applications in semantic analysis of textual documents. It can be used to develop coherent fact-based summaries of documents or to compare semantic differences between document versions. Moreover, by automatically identifying fact occurrences in documents, fact spotting can help derive valuable training data for entity linking and relation extraction tasks.

Problem Statement. Formally fact spotting is defined as follows: given a knowledge base with a set of subject-predicate-object facts F, and an arbitrary text document d, identify the subset of facts $F' \subseteq F$ that are mentioned in d. A fact $f(s, p, o)$ is considered to be spotted if s, p and o are mentioned in the document. However, in some cases one of p or o is implied and hence not stated explicitly in the text. We refer to such matches as partially spotted facts.

Fact spotting can be accomplished by mapping noun phrases in the document to the entities participating in a fact, and mapping the surrounding text to its relation. One may employ string based similarity matching to achieve these mappings. Any such similarity method needs to take into account alternative wordings for KB entities and relations; for example, "Oscar" for Academy Award, "born in" for PLACEOFBIRTH. Our earlier approach [20] uses extensive dictionaries of entity aliases and relational paraphrases for this purpose. While developing a rich yet clean dictionary of relational paraphrases is non-trivial, similarity based methods still go a long way in spotting facts.

Key Insights. Even if we had robust paraphrase dictionaries at hand, similarity based methods may still produce false positives as they match facts individually. This is because they not only overlook structure information from the document but also ignore dependencies between correlated KB facts. Following are two motivating examples:

- **Location-constrained spotting.** The sentence "`Eastwood's landmark role in Dirty Harry was critically acclaimed.`" states only one fact but has two potential matches: Eastwood –PLAYEDIN→ Dirty_Harry and Eastwood –PLAYEDROLE→ Harry_Callahan, since the textual location "`Dirty Harry`" has overlap with both the movie title and the role. However as a single textual location can mean only one entity and the movie title being the better match, the first fact should be considered spotted.
- **Spotting compound facts.** KBs such as YAGO and Freebase group together correlated facts. We refer to these groups as compound facts. An example compound fact in Freebase is $\{f_1$: Ronaldo –PLAYSFOR→ Real Madrid F. C., f_2: Ronaldo –PLAYSFROM→ 2009, f_3: Ronaldo –PLAYSATPOSITION→ Forward, f_4:Ronaldo –JERSEYNUMBER→ 7$\}$. Here f_1 is a central fact, as the facts f_2, f_3 and f_4 depend on f_1. If the evidence for f_1 in text is absent, f_2, f_3 or f_4 should not be spotted at all.

  Consider the sentence "`After winning the Champions League with Real, Ronaldo did not perform well at all in Brazil in 2014.`". The six facts that one could potentially spot here are: f_1: Ronaldo –PLAYSFORCLUB→ Real Madrid, f_2: Ronaldo –HASWON→ Champions_League, f_3: Ronaldo –WONCOMPETITIONINYEAR→ 2014, f_4: Ronaldo –PARTICIPATEDIN→ FIFA_World_Cup, f_5: FIFA_World_Cup –HAPPENEDIN→ 2014 and f_6: FIFA_World_Cup –HAPPENEDAT→ Brazil. Here f_3 is dependent on f_2, and $\{f_5, f_6\}$ are dependent on f_4 respectively, as they give temporal scopes and location for the corresponding central facts. All six are true, but the facts $\{f_5, f_6\}$ should not be spotted as they depend on the central fact f_4, and there is not really a sufficient cue on the FIFA World Cup in the sentence.

Contribution. Our solution to overcome the above issues is to take into account textual locations of the matched facts and also dependencies between KB facts. Therefore we follow a two-stage approach. We first generate a candidate set of facts using string similarity between mentions and entity labels, and between dependency paths and relational paraphrases. We then perform joint matching on the candidate set of facts to select a consistent set of KB facts by imposing constraints. For this purpose, we formulate an ILP with selection variables for textual locations, mentions and fact occurrences. Using the similarity scores as weights, the objective function maximizes the weighted set of mentions and fact occurrences, subject to two consistency constraints: i) *dependent fact constraint*: dependent facts can occur only when a central fact occurs, and ii) *textual location constraint*: a textual location can belong to at most one entity.

We evaluate our approach by spotting Freebase facts in a collection of biographies. Compared to plain similarity matching or earlier proposed methods [20], our approach delivers over 10% increase in precision with marginal drop in recall. All the evaluation results are available at `http://people.mpi-inf.mpg.de/~ttylenda/akbc2014/`

2 Related Work

Fact spotting involves both entity linking and relation identification. There exist a host of methods for matching KB entities in text, a task referred to as entity linking or entity disambiguation or record linkage (see [19] for a recent survey). Similarly, works on Wikification disambiguate mentions in input text (common nouns like *musician* as well as entity names like *Beethoven*) to their corresponding Wikipedia pages [13, 14, 11, 17]. Relation extraction techniques based on distant supervision derive training data by spotting KB entity pairs and learning reliable relational patterns [15]. While robust entity matching [10] and noise tolerant models for relation matching [18] have been the focus of study in distant supervision methods, our approach relies on joint spotting of facts. Compiling para-

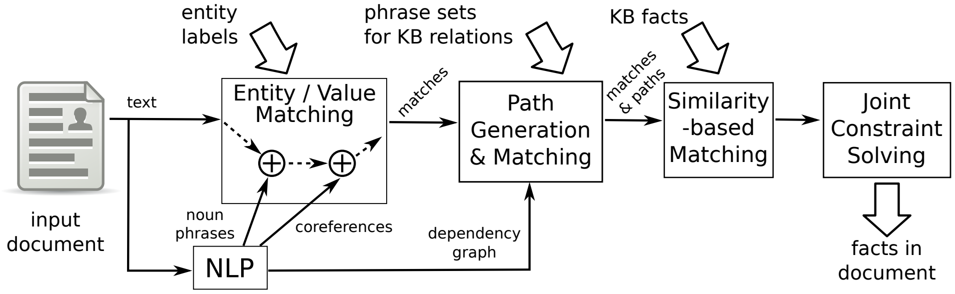

Figure 1: Overview of the fact spotting pipeline.

phrases for KB relations and entities have been studied in the context of semantic parsing [2], open question answering [5] and canonicalizing Open IE extractions [16, 7]. However, in our approach we use terms from the WordNet graph for paraphrase generation to gain higher recall.

KB provenance is related to fact spotting but provenance methods only look for corroborative evidence by analyzing the extraction sources. Joint disambiguation using ILP with constraints has been employed earlier for natural language QA over RDF graphs [21] and Wikification [4]. In this paper we use ILP with constraints to jointly disambiguate fact occurrences.

3 Methodology

KB Organization. We assume that the KB contains facts in the usual subject-predicate-object form with support for compound facts (objects can be entities, dates or values; for simplicity we will refer to all of them as entities). However, which facts are central to a compound fact needs to be manually determined, and it can easily be accomplished by marking specific relations of the compound facts as dependents. In our experiments we use facts whose relations express dates and locations as dependent facts.

Similarity-based Matching. The basic fact spotting algorithm works by matching all three constituents of an subject-predicate-object (SPO) triple to textual locations, such that the text matching the predicate connects the subject and the object. An overview of our method is presented in Fig. 1. To find entity mentions in the document we match surface form labels provided by the KB with noun phrases in the document (e.g., Victoria Caroline Adams, Posh Spice for Victoria Beckham). Surface form labels can be obtained from Freebase using /COMMON/TOPIC/ALIAS predicate, or from YAGO using the MEANS predicate. In addition to matching labels to text, we also resolve co-references using Stanford tool [12] and propagate a match at a textual location to all other locations in its coreference chain. We denote a match between an entity e and a textual location $l = (start, end)$ in the document as $match(l, e)$. Each $match(l, e)$ is associated with a weight $w(l, e)$, which captures how well the location matches the entity. These weights are obtained using Jaccard similarity over matched entity label and the text.

To match predicates in the text, we generate paths between textual locations that were matched to entities. In the simple case, all words between two locations or words within some textual window around entity mentions can constitute a path. To increase precision, however, we perform dependency parsing and use the shortest path in the dependency tree between locations. As words in the shortest path can be too restrictive, we also augment the path with dependents of those nodes. Since we propagated matches across coreference chains earlier, our method can find facts spread across multiple sentences. Note that we generate paths between locations l_1 and l_2 only if we have $match(l_1, s)$ and $match(l_2, o)$ such that there exists a fact $f(s, p, o)$ in the KB.

Matching paths directly with KB predicates does not deliver good recall. Therefore we associate with each predicate a set of phrases generated by replacing tokens in the predicate string with their neighbours in the WordNet [6] graph, specifically synsets of hyponyms, morphosemantic and derived terms. This way we generate phrases such as “romantic love affair”, “romantic friendship”, “amorous relationship” for the predicate ROMANTIC_RELATIONSHIP. Each predicate p in the KB is assigned a set of multi-token phrases E_p, which in turn have weights based on idf scores, $weight(p, phrase) = \min_{t \in phrase} idf(t)$. The intuition behind idf-based weighting is that a phrase that belongs to multiple sets is inherently ambiguous. The paths between textual locations are matched to the phrases using Jaccard similarity on token sets. For robustness, we skip stopwords and lemmatize tokens. The matching score of path against a relation is the best similarity score over its phrases, given as:

$$sim(p, path) = \max_{phrase \in E_p} \left[weight(p, phrase) \cdot Jaccard(phrase, path) \right]. \quad (1)$$

Finally, we output facts $f(s, p, o)$, where the subject s and the object o have matches in the text (some $m_{l1,s}$ and $m_{l2,o}$) which are connected by a path matching the predicate of the fact ($sim(p, path) > 0$).

Joint Constraint Solving. In order to match compound facts jointly and incorporate consistency constraints on textual locations, we develop an integer linear program (ILP), which selects a consistent set of facts from the output of the similarity-based algorithm. For matches $match(l, e)$ between textual locations and entities (or dates, values, etc.) we introduce binary variables $m_{l,e}$. The variables are set to 1 if the match is correct and 0 otherwise. Matches have weights $w_{l,e}$ coming from the full matching algorithm. They contribute $\sum_{l,e} m_{l,e} w_{l,e}$ to the objective function in ILP. To enforce that overlapping locations cannot be matched to different entities we introduce the following constraint

$$m_{l,e} + m_{l',e'} \leq 1 \text{ if } e \neq e' \text{ and } l \text{ overlaps } l'$$

We define variables f for a fact occurrence anywhere in the document, and $f_{l1,l2,path}$ for occurrence of the fact in a particular location. The variables are connected by the logical dependency $f \iff \bigvee_{l1,l2,path} f_{l1,l2,path}$, which can be translated to:

$$f \geq f_{l1,l2,path} \quad \text{for all } l_1, l_2, path, \qquad \text{and } f \leq \sum_{l1,l2,path} f_{l1,l2,path}.$$

Our constraints can "switch off" some of the overlapping location to entity matches, so we retain fully matched fact occurrences only if their subject and object are retained, i.e., $f_{l1,l2,path} \iff m_{l1,s} \wedge m_{l2,o}$ where $f = f(s, p, o)$, and $path$ between $l1$ and $l2$ matches p. In the ILP this is expressed by the constraints:

$$f_{l1,l2,path} \leq m_{l1,sub}, \qquad f_{l1,l2,path} \leq m_{l2,obj}, \qquad f_{l1,l2,path} \geq m_{l1,sub} + m_{l2,obj} - 1.$$

We associate such fact occurrence with the weight of match between the path and the predicate of the fact, so that we preferentially match facts if their predicate matches the path well. The overall objective function therefore takes the form

$$\max \sum_{l,e} m_{l,e} \cdot w_{l,e} + \sum_{l1,l2,path} f_{l1,l2,path} \cdot w_{path,p}$$

where the weight $w_{path,p}$ is given by the similarity of $path$ with predicate p as in Eq. 1. We can easily enforce the dependency between central and dependent facts $f_{dep} \implies \bigvee f_{cent}$ in the ILP as $f_{dep} \leq \sum f_{cent}$. We solve our ILP using Gurobi [8], a commercial off-the-shelf solver.

4 Evaluation

To compare the performance of our approach against [20] (referred to as WACCK) we performed fact spotting on a corpus of biographies of famous soccer players and actors collected from multiple websites. We use the ground-truth from WACCK, generated by manually annotating the biographies with Freebase facts that are mentioned therein. These annotations also contain facts whose relations are not explicitly stated in the text.

Table 1 shows the performance of a simple baseline that is often employed to generate training samples for relation extraction methods, and compares it with WACCK. The baseline approach matches only subject s and object o of a fact $f(s, p, o)$ within a sentence and assumes that the relation is present. The WACCK method improves in precision over the baseline by matching the context with relational paraphrases from the PATTY system. However, it also accepts null mappings for relations in a restricted manner, i.e., a fact $f(s, p, o)$ is considered spotted if s and o are within a prespecified distance, and f is located close to at least one other spotted fact. Nevertheless, the

			Simple baseline			**WACCK**		
Biography - Source	**GT**	**KB**	**Pr.**	**Rec.**	**F1**	**Pr.**	**Rec.**	**F1**
Arnold Schwarzenegger - msn	43	302	32.3	74.4	45.1	52.5	48.8	50.6
Clint Eastwood - msn	49	307	38.1	87.8	53.1	60.8	63.3	62.0
David Beckham - hos	41	269	38.5	97.6	55.2	55.1	65.9	60.0
David Beckham - tbc	48	269	35.6	100.0	52.5	45.5	62.5	52.6
Elizabeth Taylor - tbc	49	247	55.4	83.7	66.7	82.2	75.5	78.7
Gianluigi Buffon - hos	17	62	48.6	100.0	65.4	68.8	64.7	66.7
Jodie Foster - msn	31	204	41.3	83.9	55.3	51.9	45.2	48.3
Oliver Kahn - hos	26	71	66.7	92.3	77.4	83.3	38.5	52.6
Pelé - bio	14	108	39.1	64.3	48.6	58.3	50.0	53.8
Pelé - hos	6	108	27.3	100.0	42.9	50.0	83.3	62.5
Woody Allen - tbc	17	346	19.6	64.7	30.1	50.0	17.6	26.1
Zinedine Zidane - bio	23	139	35.5	95.7	51.8	57.1	69.6	62.7
Average (micro)	364	2432	38.8	**87.6**	53.8	58.6	58.2	58.4

Table 1: Number of facts in ground truth (GT) and Freebase (KB). Precision, recall, and F1 of baseline and WACCK methods. Biography sources: biography.com (bio), history-of-soccer.org (hos), movies.msn.com (msn), thebiographychannel.co.uk (tbc).

results in Table 1 show that higher precision can be achieved by matching relations in the input text.

Table 2 summarizes the results of our similarity-based and joint constraint solving methods. In contrast to WACCK, our similarity-based approach uses open-domain (but noisy) WordNet-based dictionary to match relations. Therefore our similarity-based method outperforms WACCK on recall, but the precision suffers. By further considering facts jointly using our ILP formulation, we obtain higher precision with a marginal fall in recall, and outperform WACCK in both these aspects. Note that our methods match all facts fully and do not consider partially stated facts at all.

			Sim.-based			**Joint**		
Biography - Source	**GT**	**KB**	**Pr.**	**Rec.**	**F1**	**Pr.**	**Rec.**	**F1**
Arnold Schwarzenegger - msn	43	302	41.3	60.5	49.1	57.5	53.5	55.4
Clint Eastwood - msn	49	307	55.4	63.3	59.0	65.2	61.2	63.2
David Beckham - hos	41	269	52.0	95.1	67.2	63.9	95.1	76.5
David Beckham - tbc	48	269	39.8	68.8	50.4	50.8	62.5	56.1
Elizabeth Taylor - tbc	49	247	75.0	67.3	71.0	85.7	61.2	71.4
Gianluigi Buffon - hos	17	62	65.2	88.2	75.0	71.4	88.2	78.9
Jodie Foster - msn	31	204	57.5	74.2	64.8	61.8	67.7	64.6
Oliver Kahn - hos	26	71	81.0	65.4	72.3	93.3	53.8	68.3
Pelé - bio	14	108	44.4	57.1	50.0	50.0	50.0	50.0
Pelé - hos	6	108	50.0	83.3	62.5	55.6	83.3	66.7
Woody Allen - tbc	17	346	32.4	64.7	43.1	50.0	35.3	41.4
Zinedine Zidane - bio	23	139	45.2	82.6	58.5	61.3	82.6	70.4
Average (micro)	364	2432	51.1	71.4	59.6	**63.4**	65.7	**64.5**

Table 2: Precision, recall, and F1 of similarity-based and joint methods.

5 Conclusion

We presented a solution to the problem of finding occurrences of known facts in previously unseen text documents. Our two-stage method matches mentions and paths to entities and relations, and then considers them jointly in an integer linear program. In comparison to exisiting approaches, joint matching helps to prune incorrectly matched facts, which improves precision without significant loss in recall. As future work, we plan to extend the joint matching to spot partially stated facts, by considering evidence from related fully spotted facts.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference (ISWC)*. Springer, 2007.

[2] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. ACL, 2013.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 2008.

[4] X. Cheng and D. Roth. Relational inference for wikification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2013.

[5] A. Fader, L. S. Zettlemoyer, and O. Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the Association for Computational Linguistics (ACL)*. ACL, 2013.

[6] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Language, speech, and communication. MIT Press, 1998.

[7] L. Galárraga, G. Heitz, K. Murphy, and F. Suchanek. Canonicalizing open knowledge bases. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*. ACM, 2014.

[8] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2014.

[9] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum. Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the International World Wide Web Conference (WWW). Companion Volume*. ACM, 2011.

[10] M. Koch, J. Gilmer, S. Soderland, and D. S. Weld. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2014.

[11] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009.

[12] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of Association for Computational Linguistics: System Demonstrations (ACL)*, 2014.

[13] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the ACM Conference on Conference on Information and Knowledge Management (CIKM)*. ACM, 2007.

[14] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*. ACM, 2008.

[15] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (EMNLP-CONLL)*. ACL, 2009.

[16] N. Nakashole, G. Weikum, and F. Suchanek. Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. ACL, 2012.

[17] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*. AACL, 2011.

[18] B. Roth, T. Barth, M. Wiegand, and D. Klakow. A survey of noise reduction methods for distant supervision. In *Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC)*. ACM, 2013.

[19] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, PP(99), 2014.

[20] T. Tylenda, Y. Wang, and G. Weikum. Spotting facts in the wild. In *Workshop on Automatic Creation and Curation of Knowledge Bases (WACCK)*, 2014.

[21] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum. Natural language questions for the web of data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. ACL, 2012.