SigmaKB: Multiple Probabilistic Knowledge Base Fusion

Miguel Rodríguez
mer@cise.ufl.edu
CISE Dept.
University of Florida
Gainesville, FL

Sean Goldberg
sean@cise.ufl.edu
CISE Dept.
University of Florida
Gainesville, FL

Daisy Zhe Wang
daisyw@cise.ufl.edu
CISE Dept.
University of Florida
Gainesville, FL

ABSTRACT

The interest in integrating web-scale knowledge bases (KBs) has intensified in the last several years. Research has focused on knowledge base completion between two KBs with complementary information, lacking any notion of uncertainty or method of handling conflicting information. We present SigmaKB, a knowledge base system that utilizes Consensus Maximization Fusion and user feedback to integrate and improve the query results of a total of 71 KBs. This paper presents the architecture and demonstration details.

1. INTRODUCTION

The amount of information available on the web has exploded and the need to corral it into a more structured form for querying, analysis, and automated reasoning has greatly increased. This has motivated a number of efforts in creating large-scale knowledge bases (KBs), each with their own methods of automatically extracting relevant information from unstructured text. This information is stored in the form of (`subject`, `relation`, `object`) triples, e.g. *Facebook*, *headquartered_in*, *Menlo_Park*. Despite sharing the same data model, each project is unique, displaying their own strengths and weaknesses related to the size of their ontology, factual completeness, method of extraction, accuracy, and domain space.

For example, the NELL [2] system continuously crawls text from the entire web and iteratively adds new facts in updates. YAGO [5] focuses primarily on heuristic extraction from semi-structured Wikipedia infoboxes. Both of these systems are tuned for high precision, which results in low coverage. On the other hand, the TAC Cold Start Knowledge Base Population (KBP) [4] and English Slot Filling (ESF) [9] tasks have motivated a varied number of small, noisy KBs with differing emphasis on precision and recall.

There have been several attempts [10, 3, 7] at *aligning* these KBs to take advantage of their complementary or potentially conflicting knowledge. They fall roughly into three categories that elucidate the difference between clean, manually constructed[1] KBs like YAGO or IMDB and noisier, automatically constructed KBs like NELL or TAC KBP. Manual-Manual alignments view the problem as one of KB completion and lack any way to deal with conflicting information. Manual-Auto and Auto-Auto alignments can probabilistically determine confidences for new facts, but efforts are limited to pairwise alignment and do not scale to aligning many KBs at once. The Linked Open Data Project[2] has done a noble job of canonicalizing and integrating many KBs, but lacks any probabilistic notion sufficient for managing opposing data.

This paper introduces SigmaKB, a probabilistic fusion system that can incorporate both strong, high-precision KBs and weaker noisy KBs into a single, cohesive master KB. SigmaKB leverages the Consensus Maximization Fusion [6] algorithm to validate, fuse, and ensemble knowledge extractions from 69 KBP knowledge bases from TAC 2015 as well as web-scale KBs such as YAGO and NELL. CM Fusion, has been shown to produce state-of-the-art results on the TAC Slot Filler Validation task[9]. SigmaKB not only provides substantial additional coverage to existing well-curated KBs, but also maintains high accuracy by ensembling across many extraction pipelines. SigmaKB generates fused confidence values to select and join queries as a unified knowledge base. In our demonstration, we show SigmaKB's query processing over 71 KBs and improvement in the quality of query results compared to state-of-the-art baselines.

2. SYSTEM OVERVIEW

SigmaKB shares the same goals as data integration systems by improving the ability to answer complex queries over multiple data sources in uncertain environments. Rather than integrate all data sources into a single, monolithic KB, we choose to remain modular, querying over each KB individually and fusing the results on-the-fly. This is similar to the LOD Cloud and increases the ease with which new KBs can be added and existing KBs can be updated. Aggregation across individual KBs is handled using a state-of-the-art Consensus Maximization Fusion framework that can leverage complementary and conflicting data values to present the user with a probabilistic interpretation of their results. Figure 1 shows the full system architecture. We describe each component in more detail below.

[1]While YAGO and DBPedia are automatically constructed, their sources are structured and manually constructed, suffering the typical problems of low coverage.
[2]http://linkeddata.org

Proceedings of the VLDB Endowment, Vol. 9, No. 13

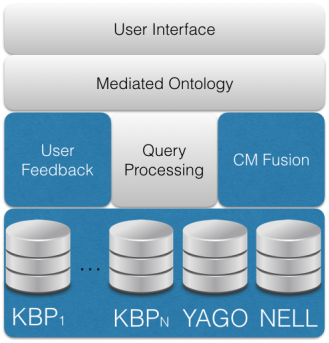

Figure 1: SigmaKB system architecture. The novel components (highlighted in blue) include incorporation of CM Fusion and user feedback over multiple KBs.

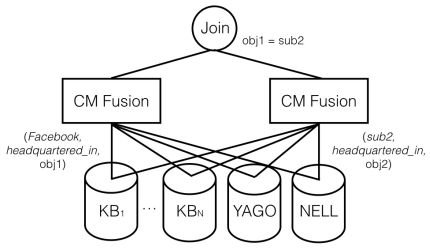

Figure 2: SigmaKB query plan for the query *Find companies headquartered in the same city as Facebook*. CM Fusion combines conflicting and complementary info before the self-join is performed.

2.1 Consensus Maximization Fusion

The key feature of SigmaKB compared to other data integration systems is the probabilistic knowledge fusion component. Rather than simply take the union of results from all individual KBs, SigmaKB contains a reasoning component that combines duplicate and conflicting entries into a cohesive, singular response returned to the user. This is especially important for some of the noisier KBs incorporated from the TAC KBP task that are prone to numerous errors, but nonetheless provide enough good information to still serve a useful purpose.

Consensus Maximization (CM) Fusion is a state-of-the-art ensemble fusion algorithm that combines multiple supervised and unsupervised classifiers. In this context, each noisy KB corresponds to an unsupervised classifier and clean, high-quality KBs act as supervised classifiers motivated by distant supervision. For a given query, CM Fusion solves a constrained optimization problem to find a canonical solution that maximizes agreement between the KBs. For each element in the union of all individual KB results, CM Fusion finds a binomial distribution over the set of true and false classes. The optimization problem is solved using a gradient descent algorithm. Figure 3 shows the pipeline for CM Fusion. Offline we collect a set of KBs that operate over multiple text sources (e.g NELL, YAGO, KBP). A preprocessing step canonicalizes string entity names and aligns different ontologies. The KBP KBs for which training data is available (from previous evaluations) are used to train 6 different meta-classifiers. The supervised and unsupervised data are passed into the consensus maximization component that produces a final aggregated probability for each triple. CM Fusion was originally applied to the TAC Slot Filler Validation task proposed by NIST where it achieved superior results compared to all other submitted systems. We refer the reader to [6] for a detailed description of the method.

2.2 User Feedback

SigmaKB has the ability to incorporate human feedback to improve all aspects of the system. This human feedback takes two forms: interface-level feedback and system-level feedback. Interface-level feedback comes from allowing users to highlight incorrect responses which are converted into negative training data for CM Fusion. System-level feedback harnesses the human computation components of the subsystems. An example comes from NELL, which adds human validated facts over time which can be incorporated as ground truth into CM Fusion. TAC KBP data is also delivered with human assertions for a small portion of queries. CM Fusion leverages the provided human feedback to bias the update procedure of gradient descent. Eventually, we anticipate adding a crowdsourcing component that can improve the results of CM Fusion by automatically querying human marketplaces such as Amazon Mechanical Turk.

2.3 Mediated Ontology & Query Processing

Knowledge bases may differ greatly in their schema, using different named and different granularity relations and properties. SigmaKB combines these different ontologies into a single mediated ontology by taking the union across all KBs and canonicalizing those relations that refer semantically to the same thing. Alignment algorithms commonly employ syntactic and structural comparisons between relations. We implemented the PARIS [8] algorithm for structure analysis, a probabilistic technique that looks at participation of `subject-object` pairs across different KBs. Syntactic analysis is achieved using the longest common substring comparisons between relations. This preprocessing step needs to be recomputed when adding a new KB with a different schema.

The query processing module uses the mediated ontology to translate from the user queries into a logical query plan across all individual KBs. Inspired by a MapReduce-like framework, we first push the translated query to each separate knowledge base before aggregating and fusing the results. An example query plan is shown in Figure 2 applied to a query for finding all the companies headquartered in the same city as Facebook.

At the individual KB-level traditional query processing techniques are applied. We currently host all KBs locally in relational databases, but future work will incorporate scalable, distributed RDF-stores as a back-end.

2.4 Interface

The user interface layer allows the user to directly submit queries in SigmaKB using SQL. We choose SQL as a query language compared to SPARQL because it allows for the expression of complex queries without adhering to a specific ontology. It also allows future incorporation of existing rela-

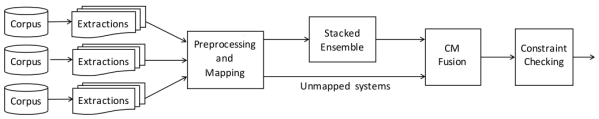

Figure 3: CM Fusion pipeline. Automatically extracted facts organized into KBs are split between labeled and unlabeled data and ensembled using Constraint Maximization.

tional databases not in RDF format. The SQL layer can be easily augmented with additional systems like SEMPRE [1] that enable natural language queries.

A couple examples of the interface are shown in Figure 4. Query results are displayed in tabular form along with provenance information. In addition to the specific knowledge bases each entry originated from, we display a unified confidence obtained using CM Fusion. Clicking on each entry brings up a tool-tip with further KB-specific info, and the ability to mark the fact as incorrect for user feedback.

3. DEMONSTRATION

3.1 Knowledge Bases

During the demonstration we showcase the ability of SigmaKB to accept SQL queries of varying degrees of complexity over a large number of KB systems. We integrated YAGO, NELL, and a set of 69 KBs submitted to the 2015 TAC KBP. To the best of our knowledge, SigmaKB will be the largest public demonstration of a usable KB fusion system. We briefly describe each KB below.

3.1.1 NELL

The Never-Ending Language Learner (NELL) acquires its facts through a number of independent modules that scrape various portions of the web through search APIs as well as a 500 million page corpus. Their primary approach uses semi-supervised NLP techniques that given a few seed examples can extract surface forms from text and use them to further improve its extraction components. At present, NELL has accumulated over 2.3 million facts over 900 types and relations.

3.1.2 YAGO

The YAGO knowledge base parses Wikipedia pages and info boxes and combines ontological information along with extensive quality control mechanisms to yield a high quality repository of facts. In our demonstration we utilize YAGO2, a more recent version that makes a number of improvements to the original design. YAGO contains 120 million facts about 10 million unique entities.

3.1.3 Knowledge Base Population

As a result of our participation in the 2015 TAC KBP Slot Filler Validation Task, we have accumulated an interesting dataset of 69 automatically extracted knowledge bases from all participating systems. They are close in spirit to OpenIE systems, but in contrast have a simple and fixed ontology. The size of these KBs is considerably smaller than that of YAGO and NELL. A key contribution of SigmaKB is the ability to augment the combination of NELL and YAGO

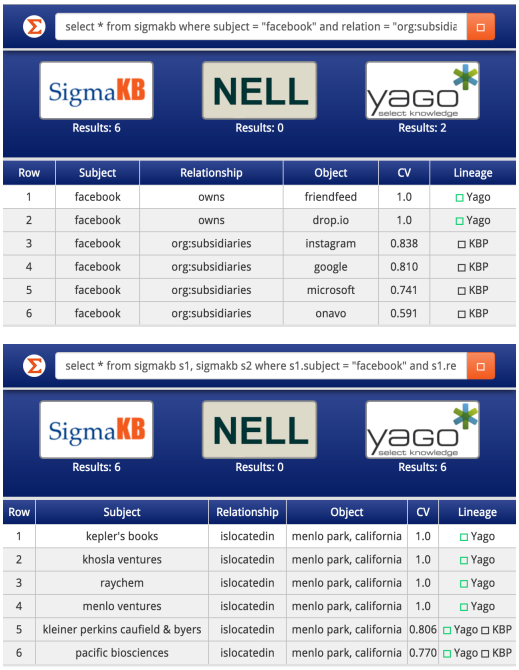

Figure 4: SigmaKB user interface. The user inputs a SQL query and results are displayed in tabular form. Clicking a row shows KB-specific provenance information. The top query lists subsidiaries of Facebook and the bottom displays companies headquartered in the same city as Facebook.

with these smaller systems that are numerous enough that they provide generous additional coverage.

3.2 Sample Queries

We now present a demonstration of several sample queries that a user may present to the system. The queries we present are part of the NIST SFV 2015 description, but our system is not limited to these queries. It can in fact answer a larger amount of queries written in SQL including but limited to aggregate queries. SigmaKB can be understood in terms of a user view as a single table with attributes `subject`, `relation`, and `object` corresponding to the triple format of each fact. Complementary information corresponds to the case where multiple facts exist with different probabilities in different KBs. Conflicting facts depend on the query, but usually pertain to two facts sharing either the same `subject` or `object` with all other values equal. We organize queries in terms of "hops". A 0-hop query corresponds to a normal `SELECT` query and a 1-hop query is a 0-hop query that uses a previous 0-hop result.

Table 1 shows results of running CM Fusion on the 2015 NIST SFV dataset containing 10,000 test queries for 0-hop and 1-hop queries. We include comparison with the top ranking individual 2015 TAC KBP system. Below we show a few specific examples of how SigmaKB can utilize CM Fusion to improve specific queries.

3.2.1 0-hop Queries

A `SELECT` query looks very much like it does in SQL, though there is very much more going on underneath the hood than a traditional RDBMS system. Consider the query

	P	R	F1	Queries
2013 & 2014	**0.528**	0.481	0.504	0-Hop
2014 only	0.477	**0.539**	**0.506**	
BBN	0.493	0.391	0.436	
2013 & 2014	**0.393**	0.097	0.155	1-Hop
2014 only	0.314	0.141	0.194	
Stanford	0.184	**0.304**	**0.229**	
2013 & 2014	**0.503**	0.307	0.381	ALL
2014 only	0.436	**0.358**	**0.393**	
BBN	0.378	0.261	0.309	

Table 1: Summary of submissions to the SFV 2015 task for different query types. We trained the meta-classifiers on 2014 ESF data or 2013 and 2014 ESF data. Comparison is made to the highest scoring individual ESF system by F1.

```
SELECT * FROM SigmaKB
WHERE subject = 'facebook' AND
relation = 'org:city_of_headquarters'
```

Though the relation is specified in terms of the KBP format, it could also be formatted in terms of YAGO's "isLocatedIn" schema. SigmaKB pushes this query to all 71 total KBs, aggregates and fuses all results to display a unified view to the observer. Our system will return a probabilistic set of results over the possible candidates the list contains candidate objects such as *San Francisco*, *Palo Alto*, *Menlo Park*, and *Chinatown* among others, with Menlo Park being the top result by a wide margin. This is because the wide consensus among multiple KB systems on *Menlo Park* greatly increases its confidence relative to other candidates. Using information that "org:city_of_headquarters" is a functional relation, we could even choose to show only the top result. If queried using a simple union, all possible candidates would be shown with their original confidences including many duplicate values of *Menlo Park*. SigmaKB shows a single unified confidence for each candidate.

It's also possible for SigmaKB to handle non-functional list-valued queries, such as the following query:

```
SELECT * FROM SigmaKB,
WHERE relation = 'org:subsidiaries'
AND subject = 'facebook'
```

Figure 4 shows the results. Since there may exist many different subsidiaries of Facebook, we define an acceptable threshold in terms of confidence and only display candidates above that value. This particular example displays the ability of SigmaKB to augment the existing large KBs YAGO and NELL. Of the 44 total results retrieved under-the-hood, only 2 originate from YAGO while the rest belong to KBP and are mostly spurious. CM Fusion, however, is able to filter to 6 higher-quality results. Top values belong to YAGO since they are of higher confidence than those from KBP. Some erroneous results did exceed the threshold such as *Google* and *Microsoft*, but this is a common problem across all KBP systems due to high co-location with Facebook in unstructured text and not a fault of SigmaKB. Overall, it shows the ability of SigmaKB to weed out conflicting information from different KBs and filter into a single, unified result.

3.2.2 1-hop Queries

SigmaKB can also handle more complex queries that require self-joins. For example, "1-hop" queries utilize the result of "0-hop" queries. An example is finding all of the companies headquartered in the same city as Facebook:

```
SELECT * FROM SigmaKB s1, SigmaKB s2
WHERE s1.subject = 'facebook' AND
s1.relation = 'org:city_of_headquarters' AND
s1.object = s2.subject AND
s2.relation = 'gpe:headquarters_in_city'
```

The results of this query are displayed in Figure 4. SigmaKB processes this query efficiently by pushing selections into the individual KB queries and performing joins on the aggregated result as evidenced in the query plan of Figure 2. This prevents us from having to process large joins between the KBs. More importantly, SigmaKB allows for information sharing between separate KBs, leveraging complementary information. Executing the above query produce actually doesn't produce any results when the KBs are queried independently. It is only through the unification of KBs in SigmaKB that we are able to respond to such queries.

4. ACKNOWLEDGMENTS

This work was partially supported by DARPA under FA8750-12-2-0348 (DEFT/CUBISM), NSF under IIS Award #1526753, a Sandia Campus Exec Fellowship and a Colciencias-Fulbright Fellowship.

5. REFERENCES

[1] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6, 2013.

[2] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI 2010*.

[3] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD*, pages 601–610, 2014.

[4] H. Ji, J. Nothman, and B. Hachey. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*, 2014.

[5] F. Mahdisoltani, J. Biega, and F. Suchanek. Yago3: A kb from multilingual wikipedias. In *CIDR 2015*.

[6] M. Rodriguez, S. Goldberg, and D. Zhe Wang. Consensus maximization fusion of probabilistic information extractors. In *NAACL 2016*.

[7] P. Shvaiko and J. Euzenat. Ontology matching: state of the art and future challenges. *Knowledge & Data Engineering, IEEE Transactions*, 25(1):158–176, 2013.

[8] F. M. Suchanek, S. Abiteboul, and P. Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *VLDB*, 5(3):157–168, Nov. 2011.

[9] M. Surdeanu and H. Ji. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*, 2014.

[10] D. Wijaya, P. P. Talukdar, and T. Mitchell. Pidgin: ontology alignment using web text as interlingua. In *CIKM 2013*.