Structuring Linked Data Search Results Using Probabilistic Soft Logic

Additional Notes on Experiments

Duhai Alshukaili

July 11, 2016

1 Introduction

The purpose of this document to provide additional details on the experiments presented in our ISWC2016 paper "Structuring Linked Data Search Results Using Probabilistic Soft Logic". The document structured as follows: Section 2 describes the dataset used in the experiments. In Section 3 provides additional notes on how we annotated the dataset with the ground truth. Last, in Section 4 we explain the evaluation metrics used in our paper.

2 Dataset

To our knowledge, there is no publicly available standard dataset that allows us to evaluate our approach. In order to learn the weights for the PSL models in described in the paper, we conducted 8 searches using Sindice [8] and Falcons [3] search engines. From each search engine, we collected the top 20 results for each term. Table 1 shows the terms used in these searches. Table 2 shows further details about the size and composition of the dataset for each domain. The most common datasets for the retuned results are DBpeida, semanticWeb.org[1], Linked Movie Database (LMDB), and MusicBrainz. While DBpedia is a general knowledge base, other sources are

[1] data.semanticweb.org

Table 1: Search terms used in constructing the evaluation dataset

Domain	**Search Terms**
Cities	Berlin,Manchester
Films	Godfather,Casablanca, Godfather actors
People	Tim Berners-Lee, Chris Bizer

Dataset	**# of Triples**	**Distinct Types**	**Distinct Properties**	**Typed Individuals**	**Common Types**		**Common Data sources**
					Type	*# of Instances*	
People	2488	42	160	70	swrc:InProceedings foaf:Document swrc:Article	30 30 9	semanticweb.org dblp.l3s.de
Films	8328	112	213	86	foaf:Person movie:actor dbo:Agent	45 33 15	linkedmdb.org dbpedia.org
Cities	16318	114	310	155	schema:MusicAlbum schema:Person pos:SpatialThing	53 19 18	musicbrainz.org dbpedia.org

Table 2: Statistical information of the evaluation dataset

domain specific. SemanticWeb.org captures information about the Semantic Web community, LMDB is online RDF database extracted from IMDB, and MusicBrainz provides information about Artists, Releases, Tracks, relationships between them. Note that while the intended domain for the terms `Berlin` and `Manchester` is *cities*, most of the returned results are not of this domain. This shows terms that are commonly considered as unambiguous, often become ambiguous as results of a wealth of WoD resources.

The results in this datasets were pre-processed by removing triples containing RDF types that belong to the *yago* and *dbyago* name-spaces. The reason is that such types (e.g, *yago:GangsterFilms* and *yago:AmericanEpicFilms*) are used to for categorizing resources as opposed to assigning real-world entity types to resources. These types can not be easily assigned specific attributes.

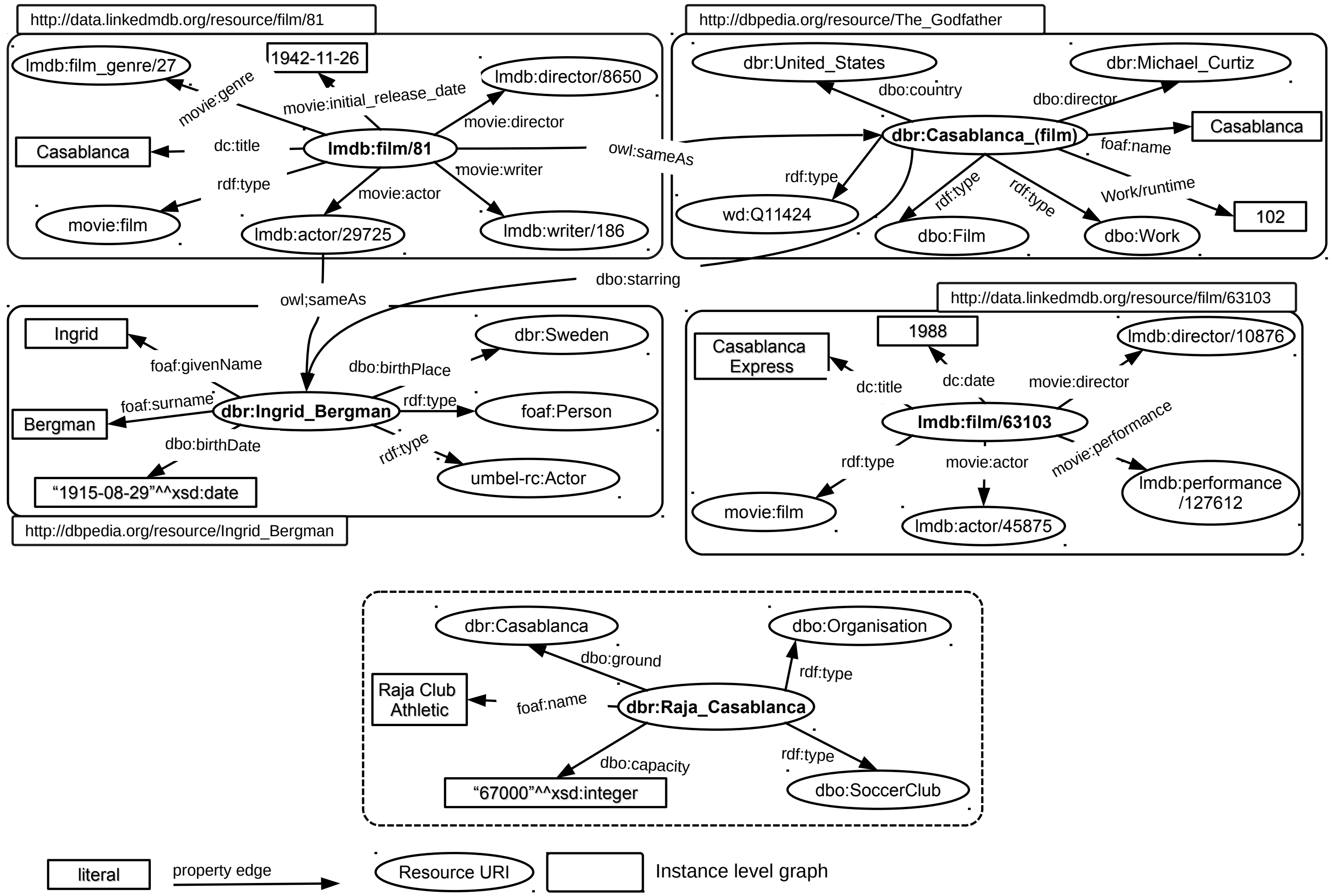

Figure 1: A RDF Sub-graph for some of the results for the term `Casablanca`

3 Ground-Truth Definition

To conduct our experimental evaluation on the collected dataset, we annotated it with the ground truth. To understand the reasoning that was adopted when annotating the dataset, we provide an example of the ground truth for some of the results of the search term `Casablanca`, shown in Figure 1. While annotating the dataset, we took into account the relevant domain of the term. For example, in *films* domain we considered, among others, the types *Film*, *Actor*, *Producer* and *Director* as true instances of *EntityType*. Conversely, the types *SoccerClub* and *Organisation* are not true instances of *EntityType* given the films domain. Similarly, the individuals *dbr:Ingrid_Bergman* and *lmdb:film/63103* (i.e., Casablanca Express) are instances of the meta type *Entity* unlike *dbr:Raja_Casablanca*. In order to define the shape of *EntityType* instances we used `schema.org` as a reference schema for the ground truth annotation of *Property* and *HasProperty*. For example, given the results in Figure 1, and with reference `schema.org`, the properties of the *EntityType* *Film* are: *actor*, *director*, *runtime*, *starring*, *country*, and *genre*. Similarly, the properties of *Actor* are: *givenName*, *surname*, *birthPlace*, and *birthDate*. Note that the properties of *Actor* in this case are also properties of the super-type *Person*. In the ground truth for our dataset, the sub-types inherit all the properties of the super-types as exemplified in here. The ground truth for the RDF graph in Figure 1 is shown in Figure 2.

4 Evaluation Metrics

To evaluate our approach, we compare the results of the MLN with the results of a random classifier. The random classifier simply assign a random number in the range $[0, 1]$ to each instances of each query predicate. The results are evaluated by computing the area under the precision-recall curve (AUC) (see Appendix A for an example on AUC computation). To compute AUC, the precision-recall (PR) curve is generated first. The precision and recall are computed according to the following:

$$Precision = \frac{\text{Number of propositions correctly predicted as positives}}{\text{Number of all propositions predicted as positives}}$$

$$Recall = \frac{\text{Number of propositions correctly predicted as positives}}{\text{Total number of postive propositions in the data}}$$

```
/* HasType GT*/
HasType(lmdb:film/81, Film)
HasType(lmdb:film/63103, Film)
HasType(dbr:Casablanca_(film), Film)
HasType(dbr:Ingrid_Bergman, Actor)
```

```
/* EntityType GT */
EntityType(Work)
EntityType(Film)
EntityType(Person)
EntityType(Actor)
```

```
/* Entity GT */
Entity(lmdb:film/81)
Entity(lmdb:film/63103)
Entity(dbr:Casablanca_(film))
Entity(dbr:Ingrid_Bergman)
```

```
/* Property GT*/
Property(initial_release_date)
Property(actor)
Property(director)
Property(writer)
Property(starring)
Property(country)
Property(runtime)
Property(genre)
Property(performance)
Property(birthPlace)
Property(birthDate)
Property(givenName)
Property(surname)
```

```
/* HasProperty GT */
HasProperty(Work, runtime)
HasProperty(Work, author)
HasProperty(Film, actor)
HasProperty(Film, starring)
HasProperty(Film, director)
HasProperty(Film, genre)
HasProperty(Film, country)
HasProperty(Film,
    initial_release_date)
HasProperty(Film, performance)
HasProperty(Person, birthDate)
HasProperty(Person, birthPlace)
HasProperty(Person, givenName)
HasProperty(Person, surname)
```

Figure 2: The ground truth annotation for the RDF in Fig. 1

A PR curve is produced by plotting a point for the precision and recall obtained at set of threshold values. The AUC is single scalar value which is a summary of the area under PR curve. This summary is often used to evaluate the performance of machine learning systems that produce continuous outputs (e.g., probability score) instead of binary ones [2]. A significant property of the AUC is the it is equivalent to the probability that classifier will rank randomly selected positive proposition higher than a randomly selected negative proposition [6].

A Computing Area Under Precision-Recall Curve

In the work presented in our paper, we use the area under precision-recall (AUC) curve to evaluate the performance of our PSL models. AUC PR is a

Query Predicate	Probabiliy	Ground Truth
HasProperty(Person, birthYear)	0.973	1
HasProperty(Person, birthDate)	0.973	1
HasProperty(Work,runtime)	0.967	1
HasProperty(Work, musicComposer)	0.811	0
HasProperty(SpatialThing, lat)	0.645	1
HasProperty(Q728937, numberOfStations)	0.645	0
HasProperty(SportsTeam, manager)	0.645	1
HasProperty(Film, producer)	0.583	1
HasProperty(Film, director)	0.573	1
HasProperty(Person, abstract)	0.44	0
HasProperty(Work, distributor)	0.368	0
HasProperty(MusicalWork, previousWork)	0.335	1
HasProperty(Location, populationMetro)	0.322	0
HasProperty(Organization, season)	0.322	0
HasProperty(Place, speedLimit)	0.322	0
HasProperty(SoccerPlayer, surname)	0.071	1

Table 3: An excerpt of ***HasProperty*** results.

summary measure that computed on the basis of two information retrieval (IR) metrics: precision and recall. Precision is a measure of result relevancy, while recall is measure of many truly relevant results are retuned. The AUC is common evaluation metric that is used by the SRL community (e.g., employed by [11, 9, 1, 7, 5, 10]. The AUC is a single point summary of resulting curve. In machine learning, the AUC us used as a heuristic for optimizing machine learning algorithms and for comparing between the performance of different classifiers [4]. The use of AUC is common in settings the involve highly skewed datasets where the number of false positives is exceeds the number of true positives. For example, in our problem there are more things that not similar (e.g., via ***SimEntity***) than things that are similar.

To compute the AUC, the PR curve need to be plotted first. This is done by varying the probability thresholds of precision and recall of a probabilistic classifier. The *threshold* (t) determines which propositions in the inference results are labelled positive and which negative. The ones whose probability of being greater than or equal the threshold are positive and the rest are negative. The *precision* (P) and *recall* (R) are computing using the standard

Threshold	0.071	0.335	0.573	0.583	0.645	0.967	0.973	*1.0*
Precision	0.562	0.667	0.778	0.750	0.714	1.0	1.0	*1.0*
Recall	1.0	0.889	0.778	0.667	0.556	0.333	0.222	*0.0*

Table 4: Obtained PR scores for the results shown in Table 3

IR formulas as follows:

$$Precision = \frac{T_P}{T_P + F_P} \qquad Recall = \frac{T_P}{T_P + F_n}$$

where T_P is the number of *true positives*, F_P is the number of *false positives*, and F_n is the number of *false negatives*.

To illustrate how the AUC is computed, consider the results shown in Table 3 which shows the some of the results produced by our PSL model for the ***HasProperty*** query predicate. Given that the distribution of probability scores varies greatly among different query predicates in the interval $[0, 1]$, the thresholds for computing the AUC are determined by the positive atoms of a query predicate. For instance, in this example, the thresholds for which the precision and recall are computed are 0.071, 0.335, 0.573, 0.583, 0.645, 0.967 and 0.973. At $t = 0.071$, $T_P = 9$, $F_P = 7$, $F_n = 0$, thus $P = 0.562$ and $R = 1$. Similarly, At $t = 0.335$, $T_P = 8$, $F_P = 4$, $F_n = 1$, so we get for $P = 0.667$ and for $R = 0.889$. Repeating this calculation for the remaining thresholds we obtain the scores shown in Table 4. These obtained scores produce the curve shown in Figure 3. The area under a curve between the upper-left and lower-right points can be found by estimating a definite integral between the two points. We use *scikit-learn* [2] tool kit to estimate the find the value of the area.

[2] scikit-learn.org/

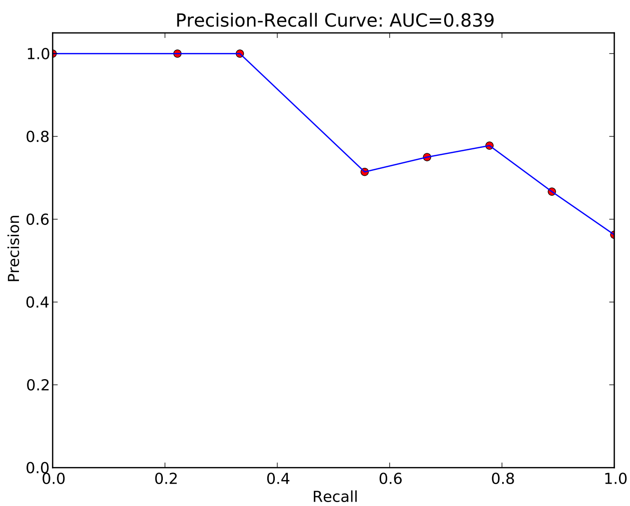

Figure 3: PR curve for the scores in Table 4

References

[1] Bach, S.H., Huang, B., London, B., Getoor, L.: Hinge-loss markov random fields: Convex inference for structured prediction. CoRR abs/1309.6813 (2013), http://arxiv.org/abs/1309.6813

[2] Boyd, K., Eng, K.H., Jr., C.D.P.: Area under the precision-recall curve: Point estimates and confidence intervals. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III. pp. 451–466 (2013), http://dx.doi.org/10.1007/978-3-642-40994-3_29

[3] Cheng, G., Qu, Y.: Searching linked objects with falcons: Approach, implementation and evaluation. International Journal on Semantic Web and Information Systems (IJSWIS) 5(3), 49–70 (2009)

[4] Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning. pp. 233–240. ACM (2006)

[5] Fakhraei, S., Foulds, J.R., Shashanka, M.V.S., Getoor, L.: Collective spammer detection in evolving multi-relational social networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015. pp. 1769–1778 (2015), http://doi.acm.org/10.1145/2783258.2788606

[6] Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27(8), 861–874 (2006), http://dx.doi.org/10.1016/j.patrec.2005.10.010

[7] Neville, J., Jensen, D.: Relational dependency networks. Journal of Machine Learning Research 8, 653–692 (2007), http://dl.acm.org/citation.cfm?id=1314522

[8] Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: a document-oriented lookup index for open linked data. IJMSO 3(1), 37–52 (2008)

[9] Poon, H., Domingos, P.M.: Sound and efficient inference with probabilistic and deterministic dependencies. In: Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA. pp. 458–463 (2006), http://www.aaai.org/Library/AAAI/2006/aaai06-073.php

[10] Pujara, J., Miao, H., Getoor, L., Cohen, W.: Knowledge graph identification. In: The Semantic Web–ISWC 2013, pp. 542–557. Springer (2013)

[11] Singla, P., Domingos, P.: Entity Resolution with Markov Logic. In: Data Mining, 2006. ICDM '06. Sixth International Conference on. pp. 572–582 (dec 2006)