Albertoni R. (2019)
Applying Predictive Models to Support skos:ExactMatch Validation.
In: Garoufallou E., Fallucchi F., William De Luca E. (eds) Metadata and Semantic Research. MTSR 2019.
https://doi.org/10.1007/978-3-030-36599-8_16

Communications in Computer and Information Science, vol 1057. Springer, Cham

Applying predictive models to support skos:ExactMatch validation

Riccardo Albertoni[1][0000-0001-5648-2713]

[1] Istituto di Matematica Applicata e Tecnologie Informatiche "Enrico Magenes", Consiglio Nazionale delle Ricerche (IMATI-CNR), Via De Marini, 6, 16149 Genova, Italy
albertoni@ge.imati.cnr.it

Abstract. The paper investigates the use of Machine Learning (ML) to support experts validating skos:exactMatch links. It trains ML techniques provided by RapidMiner with manually validated links and shows how to use the obtained predictive models for saving expert efforts. The obtained results are preliminary but encouraging: the trained predictive models reduce up to 70% the number of manual checking required from experts, leaving only 10% of the wrong links unnoticed. Cutting the 70% of the expert burden is crucial, especially when dealing with the validation of large sets of links.

Keywords: Linkset correctness, Quality, Expert validation, Predictive models.

1 Introduction

Artificial Intelligence (AI) is a multidisciplinary, long-standing research field which is recently having a hype. It promises to solve the unprogrammable tasks, to reduce the time required to program complex solutions, to make products more customizable. In particular, the advancement of Machine Learning (ML) is producing a paradigm-shift in solving problems which moves the focus from the logic of a solution to the observation of examples[1].

Following the belief that ML tools will soon become as disruptive as spreadsheets in everyday activities, this paper investigates the reuse of well-known ML technology to ease manual validation of automatically generated links. In general, the expert validation is a painstaking, error-prone and tedious activity. Any support aimed at easing the burden of validation is precious.

This paper starts from the validation carried out in the eENVPlus project (CIP-ICT-PSP No. 325232). In eENVPlus, domain experts were required to validate automatically generated *skos:exactMatch* links between the thesauri included in Linked Thesaurus Framework for the Environment (LusTRE)[1]. Distinguishing between correct and incorrect *skos:exactMatch* is particularly important in LusTRE, as user navigations and service results are enriched with translations and concepts which are reachable through these links [2], and the wrong *skos:exactMatch* links would bring to wrong enrichments. This paper trains state-of-art ML techniques made available by RapidMiner with a subset of manually validated links, and it shows how the obtained predictive models can reduce the number of manual checks required during the validation.

[1] https://developers.google.com/machine-learning/crash-course/ml-intro

2 Related Work

Link correctness is addressed by a certain number of works, most of those focus on *owl:sameAs* links. Raad et al. [3] use network metrics to check the correctness of *owl:sameAs*. CEDAL [4] provides a time-efficient method to detect inconsistent *owl:sameAs* arising from transitive closure, Papaleo et al. [5] detect logical conflicts of *owl:sameAs* links in RDF data. Paulheim [6] exploits RapidMiner multidimensional outlier detections to identifying wrong links between datasets.

Besides the works aiming at automatically identify wrong *owl:sameAs*, there are crowdsourcing-based methodologies to share the validation efforts on a larger group of experts (see [7] and [8]). None of the previous specifically address *skos:exactMatch* links, nor they learn from data of previous validations. At the best of our knowledge, Rico et al. [9] have proposed the most related approach. They exploit binary classifiers to check wrong mapping in the data extraction from Wikipedia to DBpedia. However, they rely on features which are not directly applicable to the *skos:exactMatch* links considered in this paper.

3 LusTRE and Link Validation

The eENVplus project has spent considerable efforts reviewing the available environmental thesauri and checking those not yet available as linked data [10]. As a result of such a review, we have designed LusTRE [1], in which ThiST [11] and EARTh [12] are published as Linked Data using the Simple Knowledge Organization System (SKOS) and connected to popular thesauri such as GEMET, AGROVOC [13] and EUROVOC.

LusTRE provides different kinds of links between the concepts, as the concepts belonging to separate thesauri might be equivalent (*skos:exactMatch)*, almost equivalent (*skos:closeMatch*), more specific (*skos:broadMatch*), less specific (*skos:narrowMatch*), or related (*skos:relatedMatch*).

The links are generated with a two-step procedure. Firstly, SILK [14] (http://silkframework.org/) is applied to discover new links, and then the SILK results are validated by domain experts to verify their accuracy. SILK discovers candidate links relying on user-parameterized similarity comparison. For LusTRE, a link between two concepts is added if the similarities between their preferred labels (i.e., *skos:prefLabel*), or alternative labels/synonyms (i.e., *skos:altLabel)* are greater than a given threshold. The set of discovered candidate links are then provided to experts in the form of spreadsheets (see **Fig. 1**). In the spreadsheet, each link is represented as a row with the URI of the mapped concepts (subject and object of the link are in columns *s* and *o* respectively), with their preferred labels (columns *sPrefLabel* and *oPrefLabel*), their broader (columns *sBT* and *oBT*) and related concepts (columns *sRT* and *oRT*). The 'NaN' value appears in correspondence of not available broader or related concepts; multiple broader and related terms for the same concept are separated by the symbol '|'. Concept definitions are not included as they were not available for most of the considered thesauri.

Considering the spreadsheet, the experts can catch the meaning of linked concepts, so that they can confirm if each link is a correct s*kos:exactMatch*. If needed, they can get

more information about the represented concepts resolving their links. Experts might also reject the links if wrong, or suggest to downgrade the links to another kind of matching (e.g., *skos:closeMatch*, *skos:broadMatch, skos:relatedMatch)*.

	sBT	sRT	sPrefLabel	s	o	oPrefLabel	oBT	oRT
61	analysis	trace-element analyses| standard rocks| flame ...	quantitative analysis	../ThIST/ analisiquantitativa	http://eurovoc.europa.eu/6272	quantitative analysis	research method	NaN
62	Arctic region| Denmark	Laurentia| glacial rebound| Atlantic Ocean	Greenland	../ThIST/gronland	http://eurovoc.europa.eu/1188	Greenland	Nordic Council countries| North America	regions of Denmark
63	Asia	NaN	Far East	../ThIST/estremooriente	http://eurovoc.europa.eu/956	Far East	NaN	NaN
64	Asia	Gobi Desert	Mongolia	../ThIST/mongolus	http://eurovoc.europa.eu/1968	Mongolia	Far East	NaN
65	Asia	Indian Ocean| Indian Peninsula| Indian Shield	Sri Lanka	.../ThIST/srilanka	http://eurovoc.europa.eu/4246	Sri Lanka	SAARC countries| South Asia	NaN

Fig. 1. Excerpt of the validation spreadsheet proposed to the experts

The following section introduces how we applied the ML techniques to distinguish between correct and incorrect *skos:exactMatch*. In particular, we consider the set of links from ThIST to EARTh, Agrovoc, DBpedia, Eurovoc, which overall includes 4236 links.

4 Methods and Experiment Setting

We model the task of distinguishing between correct and incorrect *skos:exactMatch* as a binary classification. Our binary classification labels links into two classes: **Exact-Match** indicating the correct *skos:exactMatch* and **Not ExactMatch** which includes the erroneous links as well as all the other SKOS mappings (e.g., *skos:closeMatch, skos:broaderMatch)*.
To train the classifier, we need to select a set of features characterizing the links. As discussed, experts have assessed the correctness of links relying on their knowledge and also the annotations provided in the spreadsheet. The expert knowledge is not easily representable, but we can elaborate on the annotations (i.e., preferred label, broader and related terms) to compare the context in which the concepts are defined.

4.1 Features

We consider three types of features: the presence of the annotations, text similarity and composed text similarity applied to the annotations.
We deploy two text similarity metrics: the *nhammingSim*, which is the hamming normalized similarity available in the package textdistance[2], and *wmdistance*, which is the cosine on the Word2Vec embedding [15] implemented in the gensim package[3]. In particular for the latter, we use a third-party word2vec trained model[4] which includes word

[2] https://pypi.org/project/textdistance/
[3] https://radimrehurek.com/gensim
[4] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/

vectors for a vocabulary of 3 million words and phrases trained on roughly 100 billion words from a Google News dataset.
As observable in **Fig. 1**, many of the concepts involved in the links have multiple broader and related concepts (e.g., *Arctic region* and *Denmark* in row 62), and their preferred labels, their broader and related terms are often composed labels (e.g., *Arctic region, glacial rebound*). To deal with multiple and composed labels, we split and flatten them in sets of single words and we apply specific functions[5]. Given two sets of single words indicated as *X* and *Y* (e.g., for the subject's broader in row 62, X={Arctic, Region, Denmark}) and one of the text-similarity metrics indicated as *sim*, we define the functions (1) and (2).

$$Max(X, Y, sim) = max_{i,j}(sim(x_i, y_j)) \quad (1)$$

$$SummingMax(X, Y, sim) = \sum_i \ max_j \ (sim(x_i, y_j)) \quad (2)$$

The function *Max* (1) returns the maximum pairwise similarity. The function *SummingMax* (2) sums the maximum similarity for each x_i.
In this way, we get 8 features for each kind of annotation. For example, considering broader terms, we have: *sBT_missing* and *oBT_missing* which are true when the subject and the object broader terms are not available; *BT_wmdistance, BT_Mwmdistance, BT_SMwmdistance,* which are obtained comparing the concepts' broader terms by applying *wmdistance* directly and in the functions *Max* and *SummingMax; BT_nhammingSim, BT_MnhammingSim, BT_SMnhammingSim* which are obtained by applying *nhammingSim* alone and in the functions *Max* and *SummingMax*. The procedure adopted to prepare the overall 24 features (8 for preferred labels, 8 for broader terms and 8 for related terms) are available as a Jupyter Python Notebook[6].

4.2 Predictive Models and Results

This paper investigates the applicability of ML for validating *skos:exactMatch* links, not the definition of new ML algorithms. As a consequence, instead of implementing the classifiers from scratch, we decided to use the RapidMiner Framework [16]. RapidMiner is an extensible and open source ML framework which offers a collection of state-of-the-art ML algorithms. The RapidMiner Studio provides an intuitive GUI which impressively reduce the efforts required to define and compare distinct ML techniques. Exploiting RapidMiner, we build six predictive models (M1-M6), in which, we trained three classifiers (i.e., a Decision Tree for M1 and M4, a Gradient Boosted trees for M2 and M5, a Deep Learning network for M3 and M6). We use a training set that contains 172 examples out of 4236 (half **exactMatch** and half **Not exactMatch**) using all the negative examples in a 10-fold cross-validation. In the training of M1, M2, M3,

[5] This strategy works for languages such as English, Italian, Spanish which uses spaces/hyphen for dividing compound words. It may not work for German and Dutch where compound words are represented differently.
[6] https://github.com/riccardoAlbertoni/LinkCorrectness/blob/master/PreparingFeaturesForLinksetCorrectness.ipynb

we consider both the missing attributes and similarity features, for the training of M4, M5, M6, we consider only the similarity features.

	Correct ExactMatch	Wrong ExactMatch	Wrong Not ExactMatch	Correct Not ExactMatch	Precision ExactMatch	Recall ExactMatch	Precision NotExactMatch	Recall NotExactMatch	% Link to Doublecheck	% of missed not ExactMatch
M1	2899	17	1251	69	99.42%	69.86%	5.23%	80.23%	31.16%	19.77%
M2	2976	10	1174	76	99.67%	71.71%	6.08%	88.37%	29.51%	11.63%
M3	2895	14	1255	72	99.52%	69.76%	5.43%	83.72%	31.33%	16.28%
M4	3040	19	1110	67	99.38%	73.25%	5.69%	77.91%	27.79%	22.09%
M5	2924	9	1226	77	99.69%	70.46%	5.91%	89.53%	30.76%	10.47%
M6	2916	15	1234	71	99.49%	70.27%	5.44%	82.56%	30.81%	17.44%

Table 1. Classification performance.

Table 1 shows the classification performance of the models tested on the whole set of links. All the models offer very good precision for **ExactMatch**, acceptable and good recall for **ExactMatch** and **Not ExactMatch**, but very low precision for **Not Exact-Match**. The models are not good enough for being used as a replacement of experts. However, we can use them for reducing the number of links the experts need to validate manually. We can ask the experts to doublecheck only the set of links classified as **Not ExactMatch** instead of the whole set of links. For example, if we apply the model M5, experts would need to focus on only the 30.76% of the initial links (i.e., 1333 instead of 4236 links) getting the chance to find the 89.53% of the wrong *skos:exactMatch*. This strategy seems quite advantageous: it would provide an error rate of about 10%, reducing the number of manual checking of 70%. Similar advantages can be obtained using the other models.

5 Conclusion

This paper shows that predictive models might ease the validation of *skos:exactMatch* correctness reducing the number of links to doublecheck manually. The results are preliminary but promising and deserve further investigations. We have not elaborated on which is the better similarity or ML technology to apply, but we have shown that even a quick and dirty approach can help to reduce the validation efforts. Each time one of the thesauri is updated in LusTRE, the links need to be revalidated, and there is room for applying the above predictive models. Considering 7 seconds on average for checking a link in LusTRE, the proposed models can make the maintainer spend two hours and half instead of more than eight hours. Perhaps, it is not a life-changing improvement but it eases the work of maintainers, and it can result extremely useful when dealing with a greater number of link. As future work, we want to investigate if other similarity measures would have worked better, if there is a minimal number of wrong and correct links to ensure acceptable performances, and to evaluate the applicability of such an approach in contexts other than LusTRE.

Acknowledgment
The author thanks RapidMiner GmbH for granting an education license of their studio

tool and the EU project eENVPlus for providing data about the validation of links.

References

1. Albertoni, R., de Martino, M., Podestà, P., Abecker, A., Wössner, R., Schnitter, K.: LusTRE: a framework of linked environmental thesauri for metadata management. Earth Sci. Informatics. (2018).
2. Albertoni, R., Martino, M. De, Podestà, P.: Quality measures for skos: ExactMatch linksets: an application to the thesaurus framework LusTRE. Data Techn. Applic. 52 (2018).
3. Raad, J., Beek, W., van Harmelen, F., Pernelle, N., Saïs, F.: Detecting Erroneous Identity Links on the Web Using Network Metrics. In: ISWC 2018, LNCS 11136. pp. 391–407. Springer International Publishing, Cham (2018).
4. Valdestilhas, A., Soru, T., Ngomo, A.-C.N.: CEDAL: time-efficient detection of erroneous links in large-scale link repositories. In: Proc. of the International Conference on Web Intelligence, Leipzig, Germany. pp. 106–113 (2017).
5. Papaleo, L., Pernelle, N., Saïs, F., Dumont, C.: Logical Detection of Invalid SameAs Statements in RDF Data. In: EKAW 2014, LNCS 8876. pp. 373–384 (2014).
6. Paulheim, H.: Identifying Wrong Links between Datasets by Multi-dimensional Outlier Detection. In: WoDOOM 2014, co-located with ESWC 2014, Anissaras/Hersonissou, Greece,. pp. 27–38 (2014).
7. Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., Lehmann, J.: Crowdsourcing Linked Data Quality Assessment. In: ISWC 2013, LNCS 8219. pp. 260–276. Springer (2013).
8. Zaveri, A., Kontokostas, D., Sherif, M.A., Bühmann, L., Morsey, M., Auer, S., Lehmann, J.: User-driven quality evaluation of DBpedia. In: I-SEMANTICS 2013, Graz, Austria, September 4-6, 2013. pp. 97–104. ACM (2013).
9. Rico, M., Mihindukulasooriya, N., Kontokostas, D., Paulheim, H., Hellmann, S., Gómez-Pérez, A.: Predicting incorrect mappings. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18. pp. 323–330. ACM Press, USA (2018).
10. Albertoni, R., De Martino, M., Podestà, P.: Environmental thesauri under the lens of reusability. In: EGOVIS 2014, LNCS 8465. pp. 222–236. Springer (2014).
11. Carusone, A., Olivetta, L.: Thesaurus Italiano di Scienze della Terra. Ist. Poligrafico dello Stato (2006).
12. Albertoni, R., De Martino, M., Di Franco, S., De Santis, V., Plini, P.: EARTh: An Environmental Application Reference Thesaurus in the Linked Open Data cloud. Semant. Web. 5, (2014).
13. Caracciolo, C., Stellato, A., Morshed, A., Johannsen, G., Rajbhandari, S., Jaques, Y., Keizer, J.: The AGROVOC Linked Dataset. Semant. Web. 4, 341–348 (2013).
14. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and Maintaining Links on the Web of Data. In: ISWC 2009, LNCS 5823. Springer (2009).
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. 1st Int. Conf. Learn. Represent. ICLR 2013, Scottsdale, Arizona,. (2013).
16. Hofmann, M., Klinkenberg, R.: RapidMiner: Data Mining Use Cases and Business Analytics Applications. Chapman & Hall/CRC (2013).