Using iDocument for Document Categorization in Nepomuk Social Semantic Desktop

Benjamin Adrian[1], **Martin Klinkigt**[1,2]
Heiko Maus[1], **Andreas Dengel**[1,2]
([1]Knowledge-Based Systems Group, Department of Computer Science
University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence DFKI GmbH
Trippstadter Straße 122, 67663 Kaiserslautern, Germany
firstname.lastname@dfki.de)

Abstract On the Semantic Desktop users maintain their model of the world in a formal personal information model ontology. Concepts from this ontology are used to annotate documents from desktop, allowing efficient navigation and browsing of these. However, the mental overhead required for correctly classifying new incoming document is substantial. We present the integration of the ontology-based information extraction system iDocument into the Nepomuk Semantic Desktop for classifying documents within the personal information model. A comparison is done between iDocument and the original classification system Structure Recommender. It is based on real models and documents from five Nepomuk users. Results reveal evidences that iDocument's categorization proposals are rated with higher recall and precision values and show that iDocument's result ranking corresponds to user ratings.

Key Words: semantic desktop, ontology-based information extraction, text classification, personal information model
Category: H.3.1, I.2.7, M.0, M.5, M.8

1 Introduction

On the Semantic Desktop users maintain their model of the world in a formal personal information model ontology (PIMO, cf. [SvED07]). Concepts from PIMO are used to tag documents from desktop, allowing efficient navigation and browsing of these. The Nepomuk project [GHM+07] provided a Semantic Desktop implementation based on Semantic Web techniques (RDF/S). As the mental overhead required for correctly classifying new incoming document is substantial, Nepomuk offers a component called Drop-box. Users simply drop document from their desktop into the Drop-box and get recommended instances from PIMO as possible classification candidates. The current service generating these recommendations is Structure Recommender (StrucRec, cf. [TSJB08]). For each document, StrucRec generates a flat set of instance candidates. Unfortunately, these results are neither weighted nor ordered. Users cannot change the classification behavior to for example restrict recommendations on specific parts of their PIMO (e.g., just use instance about projects for classification purpose).

Therefore, we integrate the ontology-based information extraction (OBIE) system iDocument [AMD09] into Nepomuk. iDocument generates weighted document classification proposals. It also uses extraction templates that may be written in the RDF query language SPARQL[1] to define relevant patterns of PIMO instances for categorization purpose. In order to get evidence about the quality of iDocument's functionalities, we compare both systems with data from five mid-term to long-term Nepomuk users. Each user provided ten manually classified documents and rated the quality of recommended classification proposals from both systems.

The rest of this paper is structured as follows. At first, an overview of related and previous work is given. Nepomuk and its existing recommendation facilities are illustrated in Section 3. iDocument is explained in Section 4. Next section describes the comparison between iDocument and StrucRec. Finally, we conclude comparison and evaluation results and provide an outlook of future work.

2 Related Work

Ontology-based information extraction (OBIE) systems use ontologies and incorporated instance knowledge to extract information from unstructured text. Many OBIE approaches just extend standard IE systems as done in S-Cream [HSC02] or SOBA [BCR06]. Other approaches use ontologies for extraction purpose directly. Labsky et al. use specialized forms of extraction ontologies [LSN08]. GATE has been extended with ontology gazetteers for instance recognition tasks [BTMC04]. Compared to these, iDocument bases on GATE but provides additional extraction templates. These templates define patterns that describe relevant types of instances that should be extracted from text. As iDocument expects ontologies written in RDFS, templates are specified in SPARQL. Inside the Gnowsis Semantic Desktop [SGK+06], the generation of tag recommendations was done with a system called ConTag [ASRB07]. ConTag used external Web Services for extracting named entities from documents. Privacy is an important issue, thus iDocument does not query external web services.

3 Document classification in Nepomuk

Nepomuk [GHM+07] provides a document classification component called Drop Box. Tag recommendations are generated each time the user drops a document into the Drop Box. Users may accept recommended instances as tags by clicking on them. In Nepomuk, StrucRec suggests PIMO things as tags (or document categories). StrucRec's[2] approach uses labels from PIMO instances matching

[1] see `http://www.w3.org/TR/rdf-sparql-query`
[2] please inspect `http://www.alphaworks.ibm.com/tech/galaxy` for more information

them as parts of text passages. Next, it performs a spreading activation inside the PIMO for instance disambiguation purpose. Disadvantages of StrucRec are that it does not weight its results with confidence ratios and thus does not provide any ranking facilities. It is not user adaptable as users may not select specific parts of their PIMO for document classification purpose.

4 The OBIE system iDocument

iDocument takes an RDFS domain ontology, an extraction template in SPARQL (e.g., *SELECT * WHERE {?p rdf:type foaf:Person; foaf:member ?o. ?o rdf:type foaf:Organization}*), and a document as input and finally returns an RDF model with multiple named graphs. Each graph is a possible result (called scenario) for the given template. iDocument follows a pipeline of six extraction tasks (cf. [AMD09]). (i) Normalization extracts plain text and existing meta data from an underlying text document. (ii) Segmentation partitions incoming text to segmental units i.e., paragraphs, sentences, and tokens. (iii) Symbolization recognizes matches between phrases in text and literal values of data type properties of the domain ontology. Successful matches are called symbols. (iv) Instantiation resolves recognized symbols with candidates for possible instances. (v) Contextualization resolves recognized instances, recognized object properties, and existing fact knowledge for creating fact candidates that are valid for populating a template. (vi) Population populates extraction templates with multiple variants (called scenarios) of extracted facts. Each scenario, instance, and fact is weighted with a confidence value.

5 Evaluation

The comparison between iDocument and StrucRec was done on real PIMO data. Five Nepomuk users agreed to provide their PIMO model and ten already categorized documents. For each model, we logged statistics about the amount of overall PIMO instances. In addition, we marked those instances that were used as tags for each of the ten selected documents. These tag relations were deleted from the evaluation models. Based on these evaluation models, StrucRec and iDocument generated classification proposals. The Nepomuk users received two spreadsheets with results from iDocument and StrucRec about their documents. They did not know which of the recommendation system created which result. They rated each proposals by choosing one of the following three values:

-1 The instance as such is invalid and should be deleted from PIMO. Some users had applied Nepomuk components that auto inserted noisy instances into their PIMO. Instances labeled with this value were ignored in this evaluation.

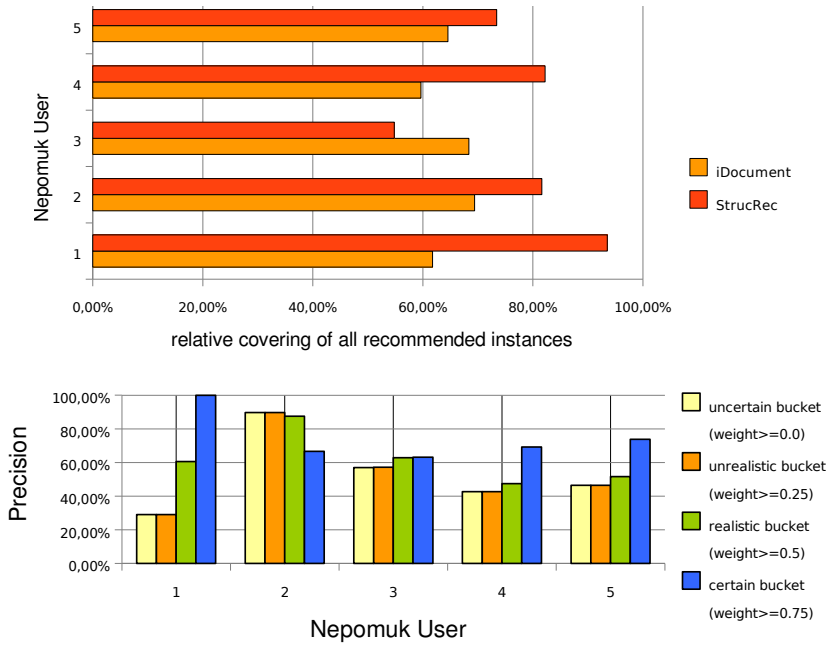

Figure 1: Upper chart: Relative amount of generated recommendations. Lower chart: Precision ratios about four ranges of iDocument's ranked result list.

0 The instance is not a valid category for this document.

+1 The instance is a valid category for this document. Users could also use this values for instances they previously did not use for categorizing documents.

If users labeled identical categories different for the same document, these categories were ignored in the evaluation. The following table shows statistics about the amount of instances each of the five users maintained in his or her PIMO.

Nepomuk User	# Instances	User Type
1	1431	(long-term user)
2	189	(mid-term user)
3	19261	(long-term user)
4	8210	(long-term user)
5	649	(mid-term user)

The upper chart in Figure 1 shows the relative covering of recommended instances of both systems for each user. We observed that both systems generated 50% identical recommendations in general. In most cases, StrucRec generated more recommendations than iDocument.

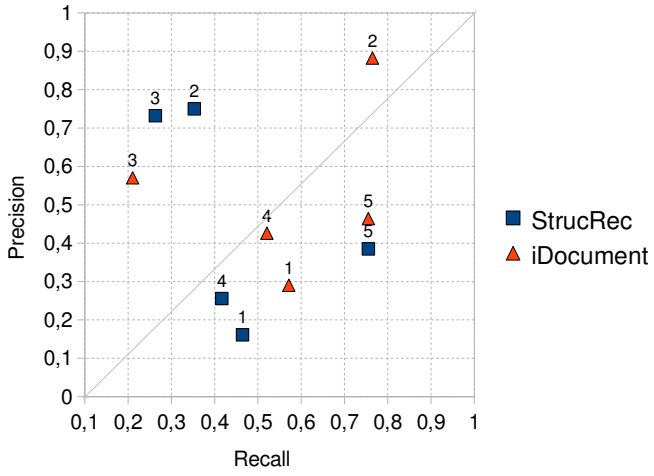

Figure 2: Recall and precision ratios

The calculation of recall was complicated as users did not know if they used all relevant PIMO instances for categorizing their documents. Thus, we could not assume users to rate over 1000 PIMO instances for each document. Therefore, we took those instances that were manually taken for categorization purpose as base line for relevant categories. We used this base line for estimating recall values about results from iDocument and StrucRec. Precision values were weighted by the amount of recommended instances each system made for each document. The distribution of precision ratios in Figure 2 shows that iDocument beats StrucRec for at least 14% except for User 3. Here StrucRec's precision of results was 14% higher than iDocument's. Analyzing recall values in Figure 2 shows that iDocument beats StrucRec for at least 10% except for User 3. Here StrucRec's recall was 5% better than iDocument's.

In contrast to StrucRec, iDocument generates confidence values for each recommendation (cf. [AD08]). We separated iDocument's recommendations into four buckets with thresholds in steps of 0.25 from 0.0 to 1.0. Each bucket contained recommended instances if the confidence was greater than the buckets threshold. Then we analyses precision ratios for each bucket. The lower chart in Figure 1 reveals that four users accepted more recommendations in buckets with higher thresholds. This confirms the quality of iDocument's result ranking.

6 Conclusion and Outlook

The comparison between StrucRec and iDocument yields that iDocument generates better recall and precision results than StrucRec for four of five users.

The user's PIMO model (User 3), where iDocument produced worse results than StrucRec, contained a huge amount of auto generated instances from other Nepomuk components. It was the largest PIMO model (19261 instances) and both systems' result were rated with poor recall values below 30%. The ranked result list of iDocument provided relevant results with high confidence weights for four of five users also. The PIMO model (User 4), where iDocument's result ranking did not fit, was the smallest of all (189 instances). As both systems generated about 50% identical recommendations, it is recommended to use both systems in Nepomuk. iDocument was executed with a standard IE template. In future work, user adaptable extraction templates are going to be evaluated. This work was financed by the BMBF project Perspecting (Grant 01IW08002).

References

[AD08] Adrian, B. and Dengel, A.: Believing finite-state cascades in knowledge-based information extraction; In *Proc. of KI 2008: Advances in Artificial Intelligence*, LNCS, pages 152–159. Springer, 2008.

[AMD09] Adrian, B., Maus, H., and Dengel, A.: The Ontology-based Information Extractions System iDocument; Poster at KM, 2009.

[ASRB07] Adrian, B., Sauermann, L., and Roth-Berghofer, T.: ConTag: A semantic tag recommendation system; In *Proc. of I-MEDIA' 07 and I-SEMANTICS' 07*, pages 297–304. JUCS, 2007.

[BCR06] Buitelaar, P., Cimiano, P., and Racioppa, S.: Ontology-based Information Extraction with SOBA; In *Proc. of LREC*, pages 2321–2324. ELRA, 2006.

[BTMC04] Bontcheva, K., Tablan, V., Maynard, D., and Cunningham, H.: Evolving GATE to Meet New Challenges in Language Engineering; *Natural Language Engineering*, 10(3/4):349—373, 2004.

[GHM+07] Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., and Gudjonsdottir, R.: The NEPOMUK Project - On the way to the Social Semantic Desktop; In *Proc. of I-MEDIA' 07 and I-SEMANTICS' 07*, pages 201–211. JUCS, 2007.

[HSC02] Handschuh, S., Staab, S., and Ciravegna, F.: S-CREAM – Semi-automatic CREAtion of Metadata; In *Proc. of SAAKM 2002 -Semantic Authoring, Annotation and Knowledge Markup, ECAI Workshop*, pages 27–34, 2002.

[LSN08] Labsky, M., Svatek, V., and Nekvasil, M.: Information Extraction Based on Extraction Ontologies: Design, Deployment and Evaluation; In *Proc. of OBIES 2008*, volume CEUR-WS 400, 2008.

[SGK+06] Sauermann, L., Grimnes, G., Kiesel, M., Fluit, C., Maus, H., Heim, D., Nadeem, D., Horak, B., and Dengel, A.: Semantic Desktop 2.0: The Gnowsis Experience; In *Proc. of ISWC*, LNCS, pages 887–900. Springer, 2006.

[SvED07] Sauermann, L., van Elst, L., and Dengel, A.: PIMO - a Framework for Representing Personal Information Models; In *Proc. of I-MEDIA' 07 and I-SEMANTICS' 07*, pages 270–277. JUCS, 2007.

[TSJB08] Troussov, A., Sogrin, M., Judge, J., and Botvich, D.: Mining socio-semantic networks using spreading activation technique; In *Proc. of I-MEDIA' 08 and I-KNOW' 08*. JUCS, 2008.