Exploiting FrameNet for Content-Based Book Recommendation

Orphée De Clercq
LT3, Language and
Translation Technology Team
Ghent University
orphee.declercq@ugent.be

Michael Schuhmacher
Research Group Data and
Web Science
University of Mannheim
michael@informatik.uni-mannheim.de

Simone Paolo Ponzetto
Research Group Data and
Web Science
University of Mannheim
simone@informatik.uni-mannheim.de

Véronique Hoste
LT3, Language and
Translation Technology Team
Ghent University
veronique.hoste@ugent.be

ABSTRACT

Adding semantic knowledge to a content-based recommender helps to better understand the items and user representations. Most recent research has focused on examining the added value of adding semantic features based on structured web data, in particular Linked Open Data (LOD). In this paper, we focus in contrast on semantic feature construction from text, by incorporating features based on semantic frames into a book recommendation classifier. To this purpose we leverage the semantic frames based on parsing the plots of the items under consideration with a state-of-the-art semantic parser. By investigating this type of semantic information, we show that these frames are also able to represent information about a particular book, but without the need of having explicitly structured data describing the books available. We reveal that exploiting frame information outperforms a basic bag-of-words approach and that especially the words relating to those frames are beneficial for classification. In a final step we compare and combine our system with the LOD features from a system leveraging DBpedia as knowledge resource. We show that both approaches yield similar results and reveal that combining semantic information from these two different sources might even be beneficial.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Content-Based Recommender Systems, Semantic Frame, Linked Data

1. INTRODUCTION

Recommender systems are omnipresent online and constitute a significant part of the marketing strategy of various companies. In recent years, a lot of advances have been made in constructing collaborative filtering systems, whereas the research on content-based recommenders had lagged somewhat behind. Similar to evolutions in information retrieval research, the focus has been more on optimizing tools and finding more sophisticated techniques leveraging for example big data than on the actual understanding or processing of the items or text at hand.

In Natural Language Processing (NLP), on the other hand, huge advances have been made in processing text both from a lexical and semantic perspective. In this respect, we believe it is important to test whether a content-based recommender system might actually benefit from plugging in more semantically enriched text features, which is the purpose of the current research. In this paper we wish to investigate to what extent leveraging semantic frame information can help in recommending books to users. We chose to work with books, since these typically contain a chronological description of certain actions or events which might be indicative for the interests of a particular reader. Someone might enjoy reading historical novels, for example, but is more prone to those novels where a love history is explained in closer detail than those where a typical revenge story is portrayed. We hypothesize that the semantic frames and or events in these two types of historical novels will be different. In other words, we wish to investigate to what extent deep semantic parsing of the plots describing a book following the FrameNet paradigm can help for recommendation.

In order to validate these claims we performed an extensive analysis on a book recommendation dataset which was provided in the framework of the 2014 ESWC challenge. What is particularly interesting about this dataset is that all the books have been mapped to their corresponding DBpedia URIs which allows us to directly compare externally

gained semantic information as available in the Linked Open Data cloud (LOD) with internal semantic information based on the plots themselves. Our analysis reveals that although some frames and events are good indicators of genres derived from external DBpedia information, they do represent some additional information which might help the recommendation process.

To actually verify this finding we test the added value of incorporating frame information as semantic features in a basic recommender system. We see that exploiting this kind of semantic information outperforms a standard bag-of-words unigram baseline and that incorporating frame elements and lexical units evoking the frames allows for the best overall performance. If we compare our best system to a system levering semantic LOD information, we observe that our frames approach is not able to outperform this system. We do find, however, that if we combine these two semantic information sources into one system we get the best overall performance. This might indicate that combining semantic information from different sources, i.e. from the linguistically grounded implicit frame features and the explicit, ontology grounded DBpedia features, is beneficial.

The remainder of this paper is structured as follows. In Section 2 we describe some related work with an explicit focus on the added value of semantic information for recommender systems. In Section 3 we then explain in closer detail the construction and reasoning behind the semantic frame-enhancement. We then continue by describing the actual experimental setup (Section 4) and have a closer analysis of the results (Section 5). We finish with some concluding remarks and ideas for future work (Section 6).

2. RELATED WORK

In content-based recommender systems, the items to be recommended are represented by a set of features based on their content, whereas a user is represented by his profile. To build a recommender both information sources are compared. Most content-based recommenders use quite simple retrieval models, such as keyword matching or the vector space model with basic TF-IDF weighting [15]. A problem with these models is that they tend to ignore semantic information. To overcome this one can use Explicit Semantic Analysis (ESA) [10] instead of TF-IDF weighting which allows to represent a document as a weighted vector of concepts. Another way to add more linguistic knowledge is to use for example information from Wordnet as done by [6, 3]. An alternative is to use language models to represent documents. This was done for example by [16] when exploring content-based filtering of calls for papers. Besides retrieval models, machine learning techniques where a system learns the user profile and classifies items as interesting or not are also used for content-based recommenders. One of the first to do this was [2] using a Naïve Bayes classifier.

When it comes to adding semantic information to recommender systems we see that currently leveraging *Linked Open Data* (LOD) is a popular research strand. [11] and [18] were among the first to use LOD for recommendation. The former use this information to build open recommender systems whereas the latter built a music recommender using collaborative filtering techniques. [4] was the first to really leverage LOD to build a content-based recommender and the first to exploit the semantics of the relations in the link hierarchy. They use LOD information from `DBpedia`, `Free-`

Table 1: Example of a frame

	Frame: KILLING	
	The KILLER or CAUSE causes the death of the VICTIM.	
FEs	KILLER	**John** drawned Martha.
	VICTIM	I saw **heretics** beheaded.
	CAUSE	**The rockslide** killed nearly half of the climbers.
	INSTRUMENT	It's difficult to suicide **with only a pocketknife**.
LUs	..., kill.v, killer.n, killing.n, lethal.a, liquidate.v, liquidation.n, liquidator.n, lynch.v, massacre.n,massacre.v, matricide.n, murder.n, murder.v, murderer.n,...	

`base` and `LinkedMDB` as the only background knowledge for a movie recommender system and show that thanks to this ontological information the quality of a standard content-based system can be improved. In more recent work, the semantic item descriptions based on LOD have been merged with positive implicit feedback in a graph-based representation to produce a hybrid top-N item recommendation algorithm, SPrank [17], which further underlines the added value of this kind of data. Moreover, in 2014 in order to spark research on LOD and content-based recommender systems, a shared task was organized by the same authors, i.e. the ESWC-14 Challenge[1].

In content-based recommendation, the advances that have been made were made possible thanks to the availability of designated datasets. These include data for predicting music, Last.FM[2], and or movies, MovieLens[3]. Up till now little research has been performed on other genres, such as books. The ESWC challenge, however, made a book recommendation dataset available which is mapped to DBpedia. DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and makes them available as linked RDF data [14]. This dataset will be used as our main data source. In this paper, we focus on the feature construction for a classifier in that we also incorporate semantic features based on the semantic frames present within the items to be recommended. This is, to our knowledge, the first approach that tries to leverage this kind of data and is one way of tackling the issue of Limited Content Analysis within recommender systems [4]. In order to validate these claims we will compare and combine our best system with a system exploiting LOD.

3. FRAME-ENHANCEMENT

In this section we give some more information about why we believe exploiting frame information might help with recommendations. First, we introduce some basic concepts and theory after which we explain how we apply a state-of-the-art semantic frame parser to our dataset and provide a first analysis. We hypothesize that a plot description tells more about a book than using more global semantic classification based on external semantic information as provided by the LOD cloud. This reasoning can be transferred to other data sources having a large number of textual information.
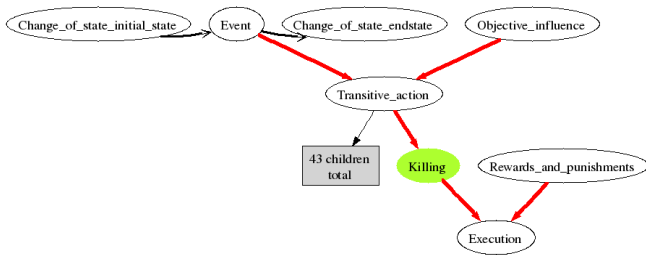

Figure 1: Example of *Inheritance* relations related to the KILLING frame.

3.1 Frame semantics and FrameNet

Following the basic assumption that the meanings of most words can best be understood on the basis of a semantic frame, FrameNet [9] was developed as a linguistic resource storing considerable information about lexical and predicate-arguments semantics in English.

FrameNet is grounded in the theory of frame semantics [7, 8]. This theory tries to describe the meaning of a sentence by characterizing the background knowledge required to understand this sentence. This knowledge is presented in an idealized, i.e. prototypical, form. A frame is thus a structured representation of a concept. It can be a description of a type of event, relation or entity, and the participants in it. In Table 1 we present an example of such a frame, *KILLING*. We see it is a semantic class containing various predicates, also known as lexical units (LUs), evoking the described situation, e.g. *killer, murder, lethal*. Moreover, it illustrates that within FrameNet each frame comes with a set of semantic roles, i.e. frame elements (FEs), which can be perceived as the participants and/or properties of a frame which are of course also lexicalized in the text itself, e.g. *Killer: John, Instrument: with only a pocketknife*.

FrameNet's latest release (1.5) contains 877 frames and about 155K exemplar sentences.[4] An interesting aspect of the FrameNet lexicon is that asymmetric frame relations can relate two frames, thus forming a complex hierarchy containing both *is-a* like and non-hierarchical relations [22]. In this work, we are particularly interested in the former type, also known as *Inheritance* relations. This type of relation entails that the child frame is a subtype of the parent frame. If we look for instance at our Killing example, of which the taxonomy is visualized in Figure 1[5], we are able to find out that this frame is a child of the frame Transitive_action, which is in turn a child of both the frame Objective_Influence and, more interestingly, the frame Event. This taxonomy thus enables us to find even more semantic properties about specific frames.

3.2 Exploiting FrameNet

3.2.1 Book dataset

For the research described in this paper, we worked with the dataset of the ESWC challenge which is in fact a re-

Table 2: Example of two sentences of a plot description and its resulting frames.

PLOT	The [Prince], the protagonist, is [named] Alexander. His [father], [Prince] Baudouin, is [**murdered**] by the [King] of Cornwall, [King] [March]. [When] Alexander [comes] of [age], he [sets out] to Camelot to [seek] justice from [King] Arthur and to [**avenge**] the [**death**] of his [father]....
FRAMES	Leadership, Appointing, Kinship, Leadership, **Killing**, Leadership, Leadership, Calendric_unit, Temporal_collocation,Arriving, Calendric_unit, Departing,Seeking_to_achieve, Leadership, **Revenge**, **Death**, Kinship.

elaborated version of the LibraryThing dataset[6]. This dataset contains books that are part of a particular user's online catalog containing the books he/she has read or owns. For the challenge, the books available in the dataset have been mapped to their corresponding DBpedia URIs [17]. Based on the available information we were able to download the plot description of each book from its corresponding Wikipedia page (this plot information is lacking in DBpedia). In this way we envisaged to investigate whether knowing more about what is actually happening in a book can enhance the recommendation. We worked with a subset by only including books of which a uniform and unambiguous DBpedia link was available and that actually contained plot information on Wikipedia. In total our final dataset contains 5,063 books with an average plot length of 312 words[7].

In order to annotate the semantic frames, each plot was parsed using the state-of-the-art frame-semantic parser SE-MAFOR [5]. This parser extracts semantic predicate-argument structures from text using a statistical model and is trained on the FrameNet 1.5 release. It takes as input the text as such, performs some preprocessing steps and outputs on a sentence-per-sentence basis all frames that are present within a text. These frames are represented by one of the 877 possible frame names and also the lexical units and frame elements (both generic and lexicalized form) are output. An example is presented in Table 2. This is the plot description of the book *The Prince and the Pilgrim*. In the text itself, the lexical units evoking the frames are indicated in square brackets. The frames and LUs which are represented in bold are those frames which actually constitute an Event. Finding out which books are events can be done by exploiting the taxonomy (cfr. supra) which enables us in a way to find out more semantic properties of specific frames. Intuitively, we can state that especially those Event frames give most information about what is happening within a book: the above-mentioned book is clearly a revenge story. However, the other frames might also pinpoint important aspects, e.g. the repetition of the Leadership and Kinship frames could inform us that this novel is about royalty and family.

What this example also illustrates is that the SEMAFOR parser is not 100% accurate. For example, the name of a particular king – King *March* – is interpreted by the parser as evoking the frame Calendric_unit. We should thus keep

---

in mind that a certain amount of noise is also introduced into our dataset. Moreover, some frames such as Arriving or Temporal_collocation, are correctly labeled but do not really contribute interesting semantic information.

For all books in our dataset we parsed the plots using SEMAFOR, after which we also filtered out those frames which can have the Event frame as a parent. Some data statistics regarding these annotations are presented in Table 3, which reveal that the information we have available is rather skewed.

Table 3: Plot annotation statistics representing the average number of real and unique frames and events per book and their standard deviations

	# Avg	Stdev	# Avg unique	Stdev
Frames	197	205	96	61
Events	42	45	22	15

3.2.2 *Semantic frames versus Linked Open Data*

As previously mentioned, we hypothesize that using frames might represent different information than using semantic information represented in the LOD-cloud. The books dataset we have at hand is particularly useful to verify this claim since all books have been mapped to their DBpedia URIs.

In order to do so, we relied on a manual subdivision of all books in genres based on LOD. This classification was made by [23] by parsing the abstract (`dbo: abstract`), the genre (`dbo:literaryGenre`, `dbp:genre`) and the subject (`dcterms: subject`) of each book against a regular expression pattern of thirty distinct genres. The authors performed this step to allow for more data coverage. However, by doing so they also made a combination of various LOD information categories which enables us to directly compare these with our semantic frames. If we have a look at our running example, *The Prince and the Pilgrim*,[8], we notice that this book is classified under the *Fantasy* genre.

Based on this genre mapping, we calculated the gain ratio [19] of our semantic frames representation with relation to the genres, thus considering the frames as features allowing to do genre classification. These gain ratios can then be observed as feature weights, and ranked according to the amount of information they add to discriminating between the thirty possible genres. We start our analysis by first only considering the semantic frame annotations. It became apparent, however, that it might be more interesting to also closer inspect those frames which are Events since these intuitively better represent what is actually happening.

The result of these analyses in presented in Table 4. Because of space constraints, we only represent the five genres representing most books of our dataset. This table each time contains the ten top features (frames and events), i.e. those with the highest gain ratio. The cell colour represents the manual analysis, indicated in light grey are those frames and events occurring only within one particular genre. In darker grey the frames and events which are representative for a specific genre are indicated. Regarding the frames, we see that it is more difficult to find distinctive features correlating with the genre (light grey). In the upper part, only the *Science Fiction* and *Crime* genre contain truly representative

[8]http://dbpedia.org/resource/The_Prince_and_the_Pilgrim

frames based on our manual analysis (dark grey). If we go to the level of the Events, we see that this already allows for finding more unique events per genre. Again, the *Science Fiction* and *Crime* genre are best represented. When we had a closer look at other discriminating features we found the same tendency. In the *Crime* genre, for example, other Events such as Verdict, Revenge, Execution, Robbery all appeared within the top twenty features.

From this analysis we could deduce that both the frames and events might deliver the same type of information as the LOD, with the events being more representative. However, what becomes clear is that the frames also contribute more information. They can represent what is happening within a book. If we again consider our running example (cfr. Table 2), which is classified as *Fantasy*, we feel that enriching a recommender with semantic frame, and especially with event information, might account for a better recommendation. This brings us to the actual experiments.

4. EXPERIMENTS

For our experiments we focus on the generation of new, semantic features. In our experimental setting we aim to evaluate the contribution of those features and thus do not explicitly focus on engineering towards a top recommendation performance.

4.1 Experimental Set-Up and Evaluation

We opt to add our semantic features to an existing recommender system [23], which participated, and performed well, in the ESWC'14 Challenge. Though we do apply feature weighting and feature selection as described below, the overall item classification and collaborative-filtering elements of the base system remain unchanged. This allows us to directly compare the predictive power of the frame-based features with the DBpedia-based features used by the original system, in particular as both approaches are different utilizations of the same information source, i.e. Wikipedia, and dataset, i.e. the ESWC RecSys Challenge data.

We use a reduced version of the dataset, based on a filtering of the 5,063 books that were retained as having sufficient plot information available (Section 3.2). This dataset has binary ratings and consists of 53,665 user-item-rating triples (6,162 users, 4,251 items) in the training data and 50,654 triples (6,180 users, 4,311 items) in the evaluation dataset.

Even though this is a binary classification task, we opt to output the positive class likelihood and not the final binary classification in order to avoid making a decision about the cut-off for the likelihood values. Consequently, we evaluate with root-mean-squared error (RMSE) to capture also the degree of confidence between the classification and the gold-standard test dataset[9]. RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - x_i)^2}$$

in which X_i is the prediction and x_i the response value, i.e. the correct value for the task at hand, and m is the number of items for which a prediction is made. Speaking in practical terms, the lower the RMSE value the better,

[9]Obtained from the ESWC'14 Challenge Chairs upon request.

16

Table 4: Top ten features with the highest gain ratios in the five most popular LOD genres. Light grey cells represent genre-unique and dark grey ones genre-representative features.

	Fantasy	Science Fiction	History	Children	Crime
FRAMES	Jury_deliberation	Beyond_compare	Representing	Memorization	Extradition
	Bond_maturation	Becoming_dry	Intentional_traversing	Measure_area	Go_into_shape
	Intentional_traversing	Containment_relation	Dominate_competitor	Estimated_value	Exporting
	Cause_to_rot	Dunking	Getting_vehicle_underway	Rope_manipulation	Becoming_dry
	Get_a_job	Exclude_member	Cause_to_rot	Degree_of_processing	Arson
	Beyond_compare	Representing	Beyond_compare	Jury_deliberation	Measure_area
	Representing	Jury_deliberation	Probability	Bond_maturation	Dominate_competitor
	Locale_by_ownership	Cause_to_rot	Jury_deliberation	Intentional_traversing	Containment_relation
	Ratification	Medium	Color_qualities	Cause_to_be_dry	Reading_aloud
	Commutation	Cause_change_of_phase	Get_a_job	Drop_in_on	Extreme_point
EVENTS	Surrendering_possession	Change_of_consistency	Eventive_affecting	Intentionally_affect	Endangering
	Dodging	Immobilization	Historic_event	Examination	Posing_as
	Immobilization	Execute_plan	Extradition	Absorb_heat	Experience_bodily_harm
	Renting	Cause_impact	Surrendering_possession	Cause_to_experience	Enforcing
	Reparation	Reparation	Corroding_caused	Fighting_activity	Cause_to_be_wet
	Heralding	Eventive_affecting	Dodging	Dodging	Intentionally_affect
	Soaking	Get_a_job	Clemency	Rope_manipulation	Intercepting
	Intentional_traversing	Cause_to_be_sharp	Intentional_traversing	Intentional_traversing	Change_resistance
	Cause_to_rot	Cause_to_rot	Cause_to_rot	Drop_in_on	Go_into_shape
	Get_a_job	Cause_change_of_phase	Get_a_job	Cause_to_be_dry	Extradition

because the closer the prediction confidence to the actual gold standard.

In addition, again motivated by wanting to avoid to choose a cut-off point for the class assignment, we follow [12] and evaluate with a receiver operating characteristic (ROC) curve and also compute the area under the curve (AUC) for it. While in contrast to RMSE, the ROC curve is computed only on the relative ordering of the predictions sorted by confidence values, it offers the advantage of understanding how a classifier would perform given different cut-off values. In addition, with ROC we can compare against recommender systems that output only an (implicit) ranking and no class confidence values.

The base system by [23] we extend is a simple content-based recommender which trains two Naïve Bayes classifiers[10] on book features acquired from DBpedia, one global classifier as background model and one per-user classifier to capture individual preferences, trained on a user-neighborhood of variable size. In our experiments, we leave this setting unchanged but only vary the different features for item representation. We experimented with five different feature representations, which is explained in closer detail in the next section.

4.2 Feature Representation

1. Baselines

First, we established two baselines: the first baseline was constructed by including the majority class based on the training data, in our case the majority class is '0'. As a second baseline we decided to include a bag-of-words approach containing token unigrams from all the different plots.

The next three groups of features all relate to the frame representation of the plots based on the SEMAFOR output (cfr. Section 3.2)

---

[10]Even though being a simple approach, in a preliminary experiment Naïve Bayes outperformed an SVM, motivating us to not compare different classifiers but focus on feature selection. In addition, Naïve Bayes was – as expected – significantly faster compared to other classifiers.

2. Frames

For the frames as such, we decided to include the resulting frame names (e.g. Killing, Kinship, Leadership) as a separate setting. In total this can lead to a maximum of 877 discriminating features, which is a large feature space shrinkage compared to the bag-of-words representation. This is why we decided to also take into consideration those particular words evoking the frames, the Lexical Units (e.g. murdered, father, Prince) on the one hand, and the lexical representations of the Frame Elements – the semantic roles – evoked by this frame on the other hand (e.g. Prince Baudouin, by the King of Cronwall, King March). In a final setting, we incrementally combine these various elements of data, thus giving more information to our classifier.

3. Events

As was illustrated in Section 3.2 the Events occurring within a book seem to intuitively represent important information of what is actually happening. This is why we also decided to perform the same experiments as with the frames but, this time only incorporating those frames which have a possible Event parent somewhere in the FrameNet hierarchy. Looking only at the Events further reduced our feature space to a maximum of 234 features. We therefore also made the same combinations as mentioned above with all possible LUs and FEs relating only to Events.

4. Taxonomy

In order to exploit the hierarchical structure of Frame-Net even further, we decided to also investigate three other settings. First we explored whether including besides a frame also its direct parent, thus going one level up in the graph, might help. We did the same in the other direction, by only including the children which are at the bottom of our taxonomy (the leafs). Another way of incorporating this graph information was to calculate for each possible frame pair that was found in a plot its least common subsumer [20] (LCS), i.e. the parent both frames have in common resulting

in the shortest path. Since the FrameNet taxonomy as such is not hypercomplex, i.e. the maximum distance between two frames is twelve, we decided to filter out those parents which are too generic by manually inspecting the LCS.[11]

For the four above-mentioned setups, the same feature selection methods were employed. Of course in order to allow for a good representation, all word-based features (bow, LUs and FEs) were first tokenized, stemmed and filtered on stop words. For the automatic feature selection, we first use unsupervised feature attribute weighting by computing the standard TF-IDF weights since all our features are in the end derived from text (book plots).

$$TF - IDF_i = \ln(1 + tf_i) \ln(N/df_i)$$

Next, we use attribute selection by computing the gain ratio with relation to the binary class label in the training data:

$$R_G(Attr, Class) = (\mathbb{H}(Class) - \mathbb{H}(Class|Attr))/\mathbb{H}(Attr)$$

This should allow us to filter out noise or unimportant features. We keep only those features with a gain ratio larger than zero ($R_G > 0$).[12]

5. Linked Open Data (LOD)

In a final setup we compare our best setting with the LOD features used by the base system, i.e. properties and values from DBpedia, and apply the same feature weighting and selection process. The features in the base system were manually selected and contain explicit book attributes, as e.g. `dbo:author` (`db:Umberto_Eco`), but also categorical information as `dbo:literaryGenre` (`db:Historical_novel`), `dcterms:subject` (`category:Novels_set_in_Italy`) or `rdf:type` (`yago:PhilosophicalNovels`) and untyped Wikipedia links in general.

We use the same set of features as reported by [23], but, to remain consistent across all experimental settings, apply our feature selection and weighting approach and use our reduced training and test dataset. In addition, we tested the combination of the DBpedia features with our best-performing frame approach.

5. RESULTS

We report experimental results for the different feature settings in Table 5. Overall, the two best performing frame features are the *Frame elements* and the *Frames+LUs+FEs*, both achieve an RMSE of 0.6036. We see that the best result is obtained when making the combination between the *Frame elements* and the LOD system, RMSE of 0.5982. Looking at the AUC for the ROC curve, both features still perform very well, but not as good as the DBpedia features alone, which achieve the best overall AUC of 0.5588.

Considering the RMSE values, we observe that the majority baseline is easily outperformed by all different settings. Looking at the bag-of-words baseline, however, illustrates that having the words of the plot available for recommendation is already a quite difficult to beat baseline.

---

[11]We looked at the most frequent LCS nodes and excluded the first 10 generic nodes such as Artifact, Relation, Intentionally_affect, Gradable_attributes, Transitive_action.

[12]Preliminary experiments revealed that keeping all features as well as doing classifier-based features selection with OneR [13] with 5-fold-cross-validation on the training data constantly underperformed against this setting.

Table 5: Experimental results on test dataset (N = 50,654) with classifier trained on different feature types (best results per category in bold).

	Features	RMSE	AUC
Baselines	Majority voting (0)	0.7705	n/a
	Words as such	**0.6145**	0.5431
Frames	Frames as such	0.6272	0.5377
	Lexical units (LUs)	0.6266	0.5398
	Frame elements (FEs)	**0.6036**	**0.5468**
	Frames + LUs	0.6259	0.5389
	Frames + LUs + FEs	**0.6036**	0.5453
Events	Events as such	**0.6132**	0.5148
	Events + LUs	0.6259	**0.5310**
	Events + LUs + FEs	0.6237	0.5296
Taxonomy	Frames One up	**0.6244**	0.5297
	Frames Bottom	0.6253	0.5370
	Frames + LCS	0.6285	**0.5376**
LOD	DBpedia features	0.6022	**0.5588**
	DBpedia + FEs	**0.5982**	0.5498
	(DBpedia + FEs hybrid)	(0.5664)	(0.5571)

Contrary to our expectations, our settings with only frames or events do not outperform this baseline. We do see that the events as such, which constitute a much smaller feature space, perform slightly better than the frames. The bag-of-words baseline is only outperformed when using features actually presenting some sort of word filtering mechanism: the Frame Elements are the lexical representation of words which are evoked by certain frames in the form of semantic roles. Even though these features are extracted from the text, it performs better than the bag-of-words *(Words as such)* baseline approach (0.6145) which does not make use of any semantic information. Analyzing the R_G-ranked feature attributes revealed that also for the other best frame approach *Frames+LUs+FEs*, the dominant attributes are the *Frame elements*, these were ranked highest. What is strange is that we do not find a similar trend when performing the same combination with our Event frames. This is probably because the feature space is too small to make a well-informed decision.

Figure 2 presents the ROC curves for our features, for the sake of readability only the most interesting curves are plotted. As to be expected from the AUC values, all curves are very close together. Besides not being far away from the diagonal, for no curve a clear cut-off value is recognizable. We observe that the DBpedia features are slightly better for the left and partially the middle part of the curve, leading to the interpretation that those features are superior for recommender systems which focus on quality. Comparing the best frames-based approach (FEs) with the bag-of-words baseline (Words), we see that FEs are mostly better than just words, with some exception around a false positive rate of around 0.23.

We also compare our system with the hybrid recommender system from Ristoski at. al. [21] (AUC 0.5848), which was the second best system of the ESWC challenge and performed essentially equally well as the winning system. That system combined many different features, not only LOD, but also user ratings and explicit collaborative filtering approaches.[13]

---

[13]As that system only outputs scores for the purpose of ranking, we transformed those into confidences by dividing each
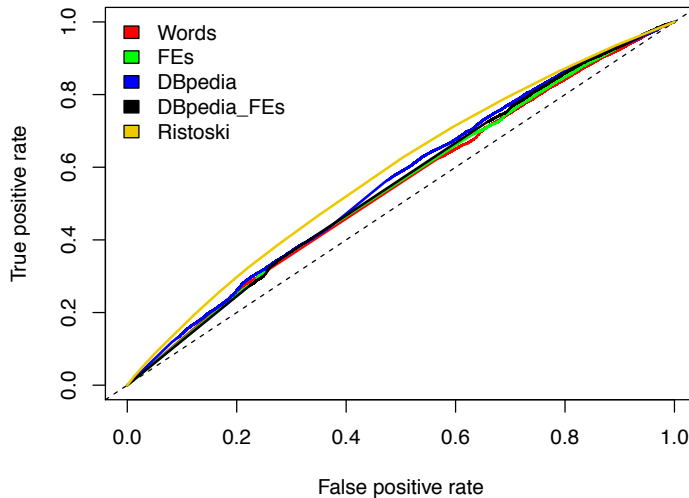
Figure 2: ROC curve for selected features and the top-performing ESWC system [21]

Looking at the ROC curve, it becomes clear that incorporating more and diverse features is beneficial in this setting. However, we have to note that this system combines different recommenders using the Borda rank aggregation method, which was not learned on the training data but manually selected while having knowledge about the test dataset (see also the comment below on our own combination model).

If we compare our best semantic frame results with the systems leveraging Linked Data, we see that we achieve a better performance (RMSE of 0.6022) when using the DBpedia features alone and that we get the best overall results when combining both our best system with the Linked Data (RMSE of 0.5982). In this way it appears that combining semantic information from different sources, i.e. from the linguistically grounded frames features and the explicit, ontology grounded DBpedia features, is beneficial in this setting. The AUC results, however, do not corroborate this finding.

Last, when not learning LOD+FEs together in one model, but separately and combine results with a simple linear combination (these results are presented in brackets in Table 5), as also done by [1], with $\lambda = 0.5$, we achieve better results (RMSE of 0.5664 and AUC of 0.5571). However, this notable improvement depends in the end on our knowledge of the test dataset, as it influenced our choice of a linear combination, instead of learning the combination of the different classifiers on the training data. Strictly speaking, this is thus not a valid experimental result, nevertheless it indicates there is most likely a better hybrid design with feature combinations that will better utilize the semantic frame features and should yield better results.

6. CONCLUSION

In this paper we have presented an alternative approach

_____

score by a constant.

to add semantic information to a content-based book recommender system. We directly compared the addition of text internal semantic frame information with text external ontological information based on *Linked Open Data* (LOD), a popular research strand.

We have shown that parsing the book plots with a state-of-the-art semantic frame parser, SEMAFOR, delivers valuable additional semantic information. This information could enable a system to fully grasp what is happening within a book. One of the added values of FrameNet is that all frames are related in a taxonomy which allows you to pinpoint those Events forming the key components of a book. Based on a direct comparison between the frames and events and a list of genres derived from DBPedia attributes, we have shown that although these data sources show some similarities, the semantic frames should be able to represent more specific information about what is happening in a particular book.

In order to test this claim in closer detail, we have performed experiments where the focus was on generating new semantic features and find out what these can contribute to a book recommendation system using one global classifier as background model and one per-user classifier. We see that exploiting semantic frame information outperforms a standard bag-of-words unigram baseline and that especially incorporating frame elements and lexical units evoking the frames allows for the best overall performance. If we compare our best system to a system levering semantic LOD information, we observe that our frames approach is not able to outperform this system. We do find, however, that if we combine these two semantic information sources into one system we get the best overall performance. This might indicate that combining semantic information from different sources, i.e. from the linguistically grounded implicit frame features and the explicit, ontology grounded DBpedia features, is beneficial.

This work has inspired many ideas for future work. Con-

sidering the current setup, we are aware that we completely relied on the output of one semantic frame parser, i.e. SE-MAFOR. We believe that using a filtering mechanism beforehand, e.g. to filter out those frames and or events which are less meaningful or noisy, or that by applying a different parser or event extraction techniques new lights can be shed on the added value of this type of information. Also, since we now only relied on Wikipedia to extract book information, we had to reduce an original larger book data. We realize a lot of additional information about books can be found online, for example on Google Books, Amazon, GoodReads, etcetera. Also the same techniques can be used to extract other types of information from both the items and users under consideration for the recommendation task.

As mentioned at the end of Section 5 we would like to further investigate whether another hybrid design might yield better results. In this respect, it would be interesting to plug our semantic knowledge in a collaborative-filtering approach to see whether this can actually help the overall performance. Using our semantic frames we could also inspect in closer detail typical problems recommender systems face such as cold-start and data sparsity.

Acknowledgments

7. REFERENCES

[1] C. Basu, H. Hirsh, W. Cohen, et al. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI'98*, pages 714–720, 1998.

[2] D. Billsus and M. J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000.

[3] M. Degemmis, P. Lops, and G. Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007.

[4] T. Di Noia, R. Mirizzi, V. Ostuni, D. Romito, and M. Zanker. Linked open data to support content-based recommender systems. In *I-SEMANTICS '12 Proceedings of the 8th International Conference on Semantic Systems*, 2012.

[5] D. Dipanjan, A. F. T. Martins, N. Schneider, and N. A. Smith. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56, 2014.

[6] M. Eirinaki, M. Vazirgiannis, and I. Varlamis. Sewep: using site semantics and a taxonomy to enhance the web personalization process. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

[7] C. J. Filmore. Frame semantics. *Linguistics in the Morning Calm*, pages 111–137, 1982.

[8] C. J. Filmore. Frames and the semantics of understanding. *Quaderni di Semantica*, IV(2), 1985.

[9] C. J. Filmore, C. R. Johnson, and M. R. L. Petruck. Background to framenet. *International Journal of Lexicography*, 16(3):235–250, 2003.

[10] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical intelligence*, 2007.

[11] B. Heitmann and C. Hayes. Using linked data to build open, collaborative recommender systems. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010.

[12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[13] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.

[14] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2013.

[15] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.

[16] G. Martín, S. Schockaert, C. Cornelis, and H. Naessens. An exploratory study on content-based filtering of call for papers. *Multidisciplinary Information Retrieval, Lecture Notes in Computer Science*, 8201:58–69, 2013.

[17] V. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *Proceedings of RecSys 2013*, 2013.

[18] A. Passant. dbrec: music recommendations using dbpedia. In *Proceedings of the 9th International Semantic Web Conference (ISWC'10)*, 2010.

[19] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[20] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence (IJCAI'95)*, 1995.

[21] P. Ristoski, E. L. Mencia, and H. Paulheim. A Hybrid Multi-Strategy Recommender System Using Linked Open Data. In *LOD-enabled Recommender Systems Challenge (ESWC 2014)*, 2014.

[22] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, and C. R. Johnson. Framenet ii: Extended theory and practice. Technical report, 2005.

[23] M. Schuhmacher and C. Meilicke. Popular Books and Linked Data: Some Results for the ESWC-14 RecSys Challenge. In *LOD-enabled Recommender Systems Challenge (ESWC 2014)*, 2014.