Bag-of-Entities text representation for client-side (video) recommender systems

Jaroslav Kuchař[*]
Web Engineering Group
Faculty of Information Technology, Czech Technical University, Prague
jaroslav.kuchar@fit.cvut.cz

Tomáš Kliegr[†]
Department of Information and Knowledge Engineering
University of Economics, Prague
tomas.kliegr@vse.cz

ABSTRACT

Client-side execution of a recommender system requires enrichment of the content delivered to the user with a list of potentially related content. A possible bottleneck for client-side recommendation is the data volume entailed by transferring the feature set describing each content item to the client, and the computational resources needed to process this feature set. This paper investigates whether the representation of the textual content (e.g. of videos) with Bag of Entities (BoE) vector generated by a wikifier can yield a classifier with the same accuracy at smaller size than the standard BoW approach. Experimental evaluation performed on the Reuters-21578 text categorization collection shows that there is a small improvement for small term vector sizes.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Recommender systems, TV, video, rule learning

1. INTRODUCTION

With increasing regulation and demands on user data privacy running a recommender system client-side is becoming a viable option. Client-side recommendation requires enrichment of the videos delivered to the user with a list of related videos (or related content in general) selected using a content-based similarity measure.

Interest Beat (InBeat) is a generic recommender system that has been adapted to perform recommendation of related content to users of online TV. The SMART-TV use case for InBeat was first introduced at RecSys'13, since then we have extended the system with Kinect gaze tracking [7], here we present a work towards adapting InBeat to perform client-side recommendation: reranking the list of related content according to the user's profile. This approach is *privacy preserving*: the model is built solely from relevance feedback stored locally and the model building as well as execution is performed on client's hardware.

A possible bottleneck for client-side recommendation is the data volume entailed by transferring the feature set describing each video (both requested and a list of related ones) to the client, and the computational resources needed to process the feature set. This paper investigates whether the representation of videos with Bag of Entities (BoE), instead of the standard Bag of Words (BoW) could lead to classifiers with the same accuracy at smaller feature set size.

Section 2 gives an overview of the InBeat system. The experimental evaluation on Reuters-21578 collection is presented in Section 3. A demo is described in Section 4. The conclusions give a list of ideas for future work.

2. CLIENT-SIDE RECOMMENDATION

InBeat learns user profiles that are composed of rules. Apart from practical issues, such as speed and the possibility to display the user profile in an intelligible way, rules are also considered as the most expressive form of encoding preferences [3].

We have developed a Javascript implementation of the apriori algorithm for learning association rules, which can be run e.g. in any HTML5 enabled TV device. This algorithm is a pivotal part of our model building approach. The rules forming the profile can then be applied on the preselected list of related content, providing a privacy-preserving personalized recommendations.

The InBeat system has the following key features:

- Gaze tracking using Microsoft Kinect is employed to obtain implicit feedback.
- The text of the TV content is represented semantically using entities detected in the text.
- Fast and scalable infrastructure.[1]

[*]Also affiliated with the Dep. of Information and Knowledge Engineering, University of Economics, Prague

[†]Also affiliated with the Multimedia and Vision Research Group, Queen Mary, University of London

[1]InBeat obtained a runner-up award in the Recsys'13 News Recommender Challenge (`https://sites.google.com/site/newsrec2013/challenge`), which put emphasis on required to provide judgment within 100 ms.

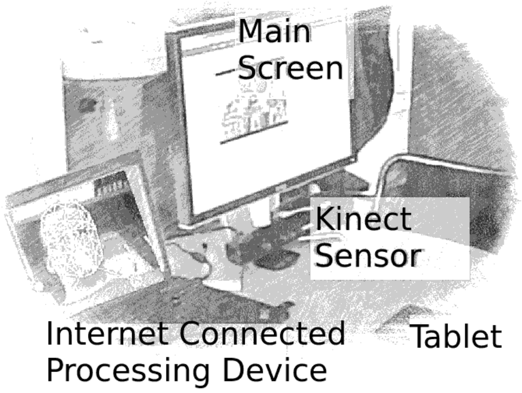

Figure 1: User-side setup

The following subsections give more detail.

2.1 User feedback with gaze tracking

The system tries to infer user interest in individual content items by observing the user's behaviour. No explicit user entry of preference information is required. There are two primary sources of feedback: player control buttons, and physical behaviour tracking. An example of user setup is given by Fig. 1.

Player control buttons: stop, pause, fast forward/ backward, skip and volume controls. Any of the listed user actions is recorded by the system and associated with the video fragment watched.

Physical behaviour tracking: is performed with Microsoft Kinect. With gaze tracking being the most important input, the system can work with other features such as child presence, which are provided by the latest version of the software library [9] that we are using to obtain high-level physical behaviour features.

Both types of feedback serve as *interest clues*. Currently, the system contains hard-coded rules such as *if the user looks at the screen, the interest is increased by 0.3*. All interest clues that have been recorded for a given content fragment contribute to the total interest score, which is then discretized to three *interest levels*: negative, neutral, positive.

2.2 Bag of Entities (BoE) representation

The input dataset consists of short textual fragments (subtitles) of the TV content being watched. Each subtitle is submitted to an entity classification service for analysis, which outputs a list of entities.

> **Example 1. (Entity annotation of text)** Consider subtitle fragment: *Luiz Felipe Scolari wants to combine the experience of the former Barcelona, AC Milan and Paris Saint-Germain star with young talent like Neymar.* The underlined word sequences are recognized as entities and disambiguated to DBpedia resources.

Our `entityclassifier.eu` component outputs, unlike most entity recognition systems, not only named entities, but also common entities (word "star" in Example 1). Each entity is assined a DBpedia URI (prefix *dbp:* in Example 2) and a list of DBpedia Ontology types (prefix *dbo:*). These types come either from DBpedia 3.9, or from the the Linked Hypernyms Dataset [6].

> **Example 2. (Entity annotation of text)** The first entity (Luiz Felipe Scolari) in Example 1 is disambiguated as `http://dbpedia.org/resource/Luiz_Felipe_Scolari`, and assigned the types: SportsManager, SoccerManager, Agent and Person.[a]
>
> [a]Using the `entityclassifier.eu` service with LHD version 2.3.9.

Each input document is then represented using a Bag of Entities (BOE). Since association rule learning cannot directly cope with weights, each entity is either present or absent from the bag. In the association rule learning terminology, each document corresponds to a *transaction*, which contains items (entities and their types). The number of times an entity or type appears in the document is ignored.

> **Example 3. (Bag of entities)** The bag of entities representation for document in Example 1:
>
> dbp:Luiz_Felipe_Scolari, dbo:SportsManager, dbo: SoccerManager, dbo:Agent, dbo:Person, dbp:Barcelona, dbo:Settlement, dbo:PopulatedPlace, ..., dbp:Talent, dbp:Experience, ..., dbp:Neymar, dbo:SoccerPlayer, ...

The motivation for using BoE instead of the bag of words (BoW) representation is that we hypothesize that this semantic web representation can help to overcome the curse of dimensionality. In cold-start recommender system problems, there is only a small number of labeled documents for which relevance feedback is available, but these contain a relatively high number of distinct words (thousands, or tens of thousand). The BoE representation a) ignores not salient words, b) uses one canonical way to represent multi word expressions that denote one "thing" (entity), and c) provides less granular features by involving entity types from the DBpedia Ontology (*dbo:* prefix).

2.3 Rule learning

For rule learning, we utilize more than twenty years of research on fast association rule learning algorithms. These are routinely used to mine massive ($p \approx 10^4, N \approx 10^8$) commercial databases [5]. A certain limitation of this learning technique is that association rule learning is not capable of processing cardinal features.

Association rule mining outputs all rules, corresponding to high density regions in the data, that comply to the preset constraints. The most commonly used constraints involve the set of permissible attributes on the left hand side (LHS) and right hand side (RHS) of the rule and a minimum value of rule interest measures. The standard measures are *confidence* and *support*.

Support is defined either as a minimum number of instances that must match the rule (absolute support) or the ratio of these instances to the total number of instances (relative support). *Confidence* can be defined as the probability that an instance will match the RHS under the condition that this instance also matches the LHS.

Example 4. (Association Rules) Considering minimum confidence threshold of 0.7 and minimum absolute support threshold of 2, examples of three association rules discovered for a given user include:

```
dbo:SoccerManager & dbp:Barcelona -> negativeInterest
dbp:Talent & dbo:SoccerPlayer -> positiveInterest
dbp:Museum -> neutralInterest
```

InBeat offers two experimental recommenders (classifiers) based on association rules: *brCBA* and *termAssoc*.

2.3.1 brCBA

The brCBA algorithm [8] is a simplified version of the seminal CBA algorithm [10]. The most important difference is that unlike CBA, it includes less or no pruning steps and outputs a partial classifier, which is simply composed of (pruned) rule set output by the association rule learner.

2.3.2 termAssoc

This setup is inspired by the ARC-BC algorithm for text categorization by term association proposed in [1].

When learning rules for a given interest level, the system takes into consideration only content fragments annotated with a given interest level. For each interest level, the system thus generates a separate list of frequent itemsets. These frequent itemsets are converted to rules predicting the current interest level.

It should be noted that while this step is inspired by the ARC-BC (Associative Rule-based Classifier By Category) algorithm [1], there are some differences. In particular, ARC-BC uses a custom apriori implementation, which redefines the support so that one transaction (document) can increase the support count by more than 1. In contrast, termAssoc relies on the "mainstream" version of the association rule learning task (with standard support definition), for which multiple performance optimizations have been proposed.

2.4 Recommendation

The recommendation is performed on a shortlist of videos, which have been found relevant to the video delivered to the user. In the InBeat demo setup, we use the ten newest related videos returned by the YouTube API for which subtitles are available .

The videos are represented by bag of entities as described in Subs. 2.2. For each unlabeled video, a rule engine identifies matching rules. A matching rule is a rule for which all entities in the LHS are contained in the BOE for the video. If there are multiple matching rules with different RHS, a conflict resolution is performed the same way as in the seminal CBA algorithm [10]: rules are sorted according to confidence (brCBA only), support, and rule length (shorter is better), and the highest confidence rule is selected. The RHS of the selected rule, which is one of the three predicted values of interest, is used to rank the videos. Since the set of learnt association rules may not cover the entire instance space, some videos may not be ranked.

3. EXPERIMENTAL EVALUATION

The focus of this section is on the evaluation of the Bag of Entities representation used in InBeat. We cast the problem as *text categorization* task. Essentially, there are three types of documents: those for which the user interest is known to be positive, negative and neutral.

The experimental setup aims at comparing the performance of the BoW representation with the BoE representation. The comparison is performed on two versions of the classifier: brCBA and termAssoc.

3.1 Dataset

We use the ModApte version of the Reuters-21578 Text Categorization Test Collection, which is one of the standard datasets for this task. The Reuters-21578 collection contains 21,578 documents, which are assigned to 135 different categories (topics). Example topics are "earn" or "wheat". One document belongs on average to 1.3 categories. We use only a subset consisting of the documents which are assigned to ten most frequently populated categories as e.g. in [1]. Our dataset thus consists of 6,399 training documents and 2,545 test documents.

3.2 Preprocessing

The preprocessing is performed in two stages. First, the BoW or BoE feature sets are created from the underlying dataset. Then, depending on the classifier used, the term (concept) vectors are pruned.

3.2.1 BOW

The input documents contain 58,714 of distinct terms. To decrease the dimensionality, we performed the following operations: all terms were converted to lower case, numbers were removed, punctuation was removed, stop words were removed[2], whitespace was stripped and the documents were stemmed. The final document-term matrix contained 25,604 terms.

3.2.2 BOE

The `entityclassifier.eu` [4] was used to wikify the documents. The web service returned a list of entities (identified as DBpedia resources) and for each entity a list of the types (DBpedia Ontology concepts).

The result of the preprocessing is a document-term matrix containing 12,878 unique concepts (entities and types).

3.2.3 Term (concept) pruning

The rule pruning is performed differently for brCBA and termAssoc algorithms. For brCBA, top N (*tvSize*) terms are selected according to TF-IDF. For termAssoc, term pruning is performed separately for each category using a TF score, selecting top N (*tvSize*) terms. Using TF-IDF scores with termAssoc degrades results in our observation, since terms with low IDF value (computed on terms within a given category) often discriminate well documents in this category w.r.t. documents in other categories.

We also tried combining the BoW and BoE representations (denoted as BoW+BoE). For a given value *tvSize* parameter, 50% were top-ranked terms from BOW and 50% top-ranked concepts from BoE.

3.3 Rule learning setup

To perform the experiments, we used the *minConf*=0.001 threshold, *minSupp*=0.001 for brCBA, and *minSupp*=0.2 for TermAssoc. The maximum rule (frequent itemset) length was unrestricted.

[2] A list of occurring 700 English stop words was used.

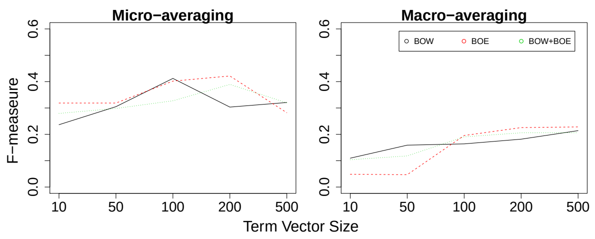

Figure 2: Results – brCBA

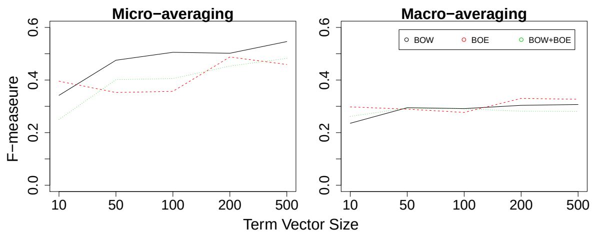

Figure 3: Results – termAssoc

3.4 Results

The results are reported in terms of micro-average and macro-average F-measure (refer to [2] for details).

The results, depicted on Fig. 2- 3, indicate that for the smallest term vector size, BoE representation yields better overall F-Measure than the BoW representation. Also, the best overall results are provided by the termAssoc algorithm.

Surprisingly, the fusion of BoW and BoE into one term vector is dominated by the performance of the BoW/BoE alone.

It should be noted that significantly better results than we have achieved on Reuters-21578 are reported e.g. in [2], also the relative improvement provided by the BoE representation is only marginal, pointing at the need to perform more research into generating the BoE feature set.

4. DEMO

The demo (available at `http://inbeat.eu/demo/`) captures user interaction with a modified YouTube player. Several videos forming a news block were preselected. It is possible to define custom set of YouTube videos (English subtitles are mandatory). Entities are automatically recognized in the subtitles and shown next to the video. The user feedback is obtained using both explicit (remote control operation) and implicit (gaze) action tracking. Each subtitle corresponds to one training document. The class label (not interesting, neutral, interesting) is obtained by aggregating the user feedback obtained through the duration of the video segment relating to the subtitle.

After sufficient number of training examples has been gathered, rule learning can be started. The demo shows a list of the resulting preference rules. These preference rules are used to rank candidate content (related YouTube videos) for personalized recommendation.

5. CONCLUSION

An important performance factor when devising client-side recommender systems is the size of the feature set. This paper presented an experimental validation of the Bag of Entities (BoE) representation on a standard dataset. Our hypothesis was that the BoE representation, obtained by running the textual transcript of the videos through a wikifier, provides better accuracy at a given term vector size than the standardly used Bag of Words (BoW). Experimental evidence obtained on Reuters-21578 text categorization collection suggests that the BoE representation can yield indeed slightly better results (F-Measure) with very small term vector size, although the increase is not as large as we have hoped for.

We also presented a demo of a proof of concept system for client-side recommendation focusing on obtaining relevance feedback with gaze tracking. The demo still leaves several server-side components in our system. As for future work, we would like to create a complete Javascript-based version of our InBeat system, which would be capable of completely standalone client side operation, and also investigate the possibilities for generating more effective feature set from the wikification output.

Acknowledgment. This research was supported by the EU via the LinkedTV project (FP7-287911), CTU in Prague grant (SGS14/104/OHK3/1T/18) and VSE by grant IGA 20/2013.

6. REFERENCES

[1] M.-L. Antonie and O. R. Zaïane. Text document categorization by term association. In *ICDM '02*, Washington, DC, USA, 2002. IEEE Computer Society.

[2] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. On feature distributional clustering for text categorization. In *SIGIR '01*, pages 146–153, New York, NY, USA, 2001. ACM.

[3] K. W. S. R. S. M. Dembczynski, K. Learning of rule ensembles for multiple attribute ranking problems. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*, pages 217–247. Springer-Verlag, 2010.

[4] M. Dojchinovski and T. Kliegr. Entityclassifier.eu: real-time classification of entities in text with Wikipedia. In *ECML'13*, pages 654–658. Springer, 2013.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[6] T. Kliegr. Towards Linked Hypernyms Dataset 2.0: complementing DBpedia with hypernym discovery and statistical type inferrence. In *LREC 2014*. ELRA.

[7] T. Kliegr and J. Kuchař. Orwellian Eye: Video recommendation with Microsoft Kinect. In *PAIS'14*. IOS Press, August 2014.

[8] T. Kliegr, J. Kuchař, D. Sottara, and S. Vojíř. Learning business rules with association rule classifiers. In *RuleML 2014*. Springer, 2014.

[9] J. Leroy, F. Rocca, M. Mancas, and B. Gosselin. 3d head pose estimation for TV setups. In *Intelligent Technologies for Interactive Entertainment*, volume 124, pages 55–64. Springer, 2013.

[10] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD'98*, pages 80–86, 1998.