Personalized News Event Retrieval for Small Talk in Social Dialog Systems

Lucas Bechberger[★△], *Maria Schmidt*[★], *Alex Waibel*[★], *Marcello Federico*[△]

[★]Interactive Systems Lab (ISL), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe
[△]Human Language Technology (HLT), Fondazione Bruno Kessler (FBK), Trento (Italy)
Email: lbechberger@uos.de, {maria.schmidt,waibel}@kit.edu, federico@fbk.eu
Web: http://isl.anthropomatik.kit.edu, http://hlt-mt.fbk.eu

Abstract

This paper presents the NewsTeller system which retrieves a news event based on a user query and the user's general interests. It can be used by a social dialog system to initiate news-related small talk.

The NewsTeller system is implemented as a pipeline with four stages: After collecting a large set of potentially relevant news events, a classifier is used to filter out malformed events. The remaining events are then ranked according to a relevance value predicted by a regressor. In a final step, a short summary of the highest-ranked event is generated and returned to the user.

Both the classifier and the regressor were evaluated on hand-labeled data sets. In addition to this, a user study was conducted to further validate the system. Evaluation results indicate that the proposed approach performs significantly better than a random baseline.

1 Introduction

This paper is concerned with the field of social dialog systems – dialog systems capable of non-task-oriented behavior serving mainly social purposes. More specifically, it explores the area of news-related small talk, i.e., small talk referring to recent news events.

Reeves & Nass [1] argue that humans treat computer systems as social actors and expect their behavior to be in line with social norms. If a computer system (e.g., a humanoid household robot) fails to fulfill its social role, users tend to perceive it as incompetent and impolite. In order to be well accepted by their users, systems that use spoken dialog as a modality will need to fulfill their role as social actors. This "social competence" includes (among other things) the ability to conduct small talk. Since many small talk conversations start with remarks about recent news events, this paper puts its focus on the specific area of news-related small talk.

This paper introduces the NewsTeller system which retrieves a news event based on a user query and the user's general interests (both represented as list of keywords, e.g., $\{Obama, Rome\}$). It can be used by a social dialog system to initiate news-related small talk by responding to a user utterance with a relevant news event.

The remainder of this paper is structured as follows: Section 2 presents related work, Section 3 describes the NewsTeller system in more detail, Section 4 analyzes the results of a user study and Section 5 concludes the paper.

2 Related Work

The NewsTeller system presented in this paper was designed to be used as a module by the social dialog system developed at ISL [2] in order to facilitate the initiation of small talk. In contrast to traditional architectures of spoken and text-based dialog systems (see e.g., [3, 4]), the dialog management component of this system is not monolithic. Instead, it consists of a set of modules that focus on different aspects of the dialog (e.g., answering factoid questions, or generating inquiring questions). These modules are executed in parallel, their responses are collected, and the response with the highest confidence is chosen as system output. This modular structure makes the system easily extensible. The NewsTeller system can be thought of as an additional module of this social dialog system: Based on keywords from the user utterance, the NewsTeller system can try to retrieve a relevant news event. Its response would then compete with the responses of other modules.

The events being processed by the NewsTeller system have been obtained by using the NewsReader NLP (Natural Language Processing) pipeline [5, 6]. This NLP pipeline processes massive amounts of online news articles and automatically extracts events from them. These events contain information about *what* happend *when* and *where*, and *who* was involved (e.g., ⟨*visit – 03/11/2016 – Rome – Barack Obama*⟩). The KnowledgeStore [7] subsystem serves as a central repository for all modules in this NLP pipeline. It stores both the original news articles and the extracted events, entities, and relationships. This latter abstract representation is stored in the form of an RDF (Resource Description Format) graph. A KnowledgeStore instance based on news articles from Wikinews was used as main data source for the development of the NewsTeller system.[1]

For ranking the events according to their predicted relevance, the "learning to rank" paradigm from information retrieval is used. Liu [8] gives a thorough overview of this topic. In "learning to rank" approaches, the ranking problem of information retrieval applications (sorting documents according to their expected relevance to a given query) is mapped to a machine learning problem.

In this paper, we focus on a specific mapping: using a regression-based approach with multiple ordered categories. In this case, the data points are labeled using a number of different classes with increasing value, e.g., "bad"(0), "fair"(1), "good"(2), "excellent"(3), and "perfect"(4) [9]. Then, a regressor is trained on these class labels (often using a modified value of $2^{classLabel} - 1$ to emphasize more relevant documents). One of the common regression models used in this setting are random forests. A random forest is an ensemble of decision trees that were trained on random subsets of the data set, using a random subset of the available features. Mohan et al. [10] showed that simple random forests used "off the shelf" without modifications can compete with more complex state of the art systems (like gradient boosted regression trees).

[1]This KnowledgeStore instance is freely accessible online at http://knowledgestore2.fbk.eu/nwr/wikinews/ui

The NewsTeller system is of course not the first contribution to dialog systems with news-related functionality.

For instance, Yoshino et al. [11–13] present a spoken dialog system for news navigation based on a partially observable Markov decision process (POMDP) and predicate-argument structures. Their system is capable of presenting news stories to the user (reading out the news headline), summarizing them (using the lead sentences of the article) and answering questions about the news stories. Although the system of Yoshino et al. does not explicitly focus on small talk, it is still strongly related to the work presented in this paper as it considers an information-seeking dialog in the news domain. However, their system does not take into account the user's general interests at all.

Within the research area of recommendation systems, some approaches focus on the recommendation of news articles, e.g., in a digital newspaper application. Both Diaz et al. [14, 15] and Billsus & Pazzani [16] present such a system. Billsus & Pazzani use a naive Bayes classifier in combination with a "nearest neighbor" approach for selecting relevant articles, whereas Diaz et al. compare candidate articles to the user model by using the cosine similarity of tf-idf vectors. Both systems divide their user model into short-term and long-term interests, and the goal of both approaches is to present a list of news articles to the user. The NewsTeller system, however, aims to present only a single news event. Moreover, a user query is taken into account in addition to a user model.

3 Approach

3.1 Overall Architecture

Figure 1 shows the overall architecture of the NewsTeller system. It uses the user query and the user model provided by the social dialog system to find a relevant news event. This is done by using the contents of the KnowledgeStore (which has been populated by the NewsReader NLP pipeline). The NewsTeller system returns a short summary of the selected news event to the social dialog system. The user model used for this project is relatively simple: It consists of a bag of weighted keywords describing the user's general long-term interest and a list of events that have been presented to the user in earlier turns.

The workflow of the NewsTeller system can be broken down into four steps that form a pipeline:

- **Event Search**: fetches a large number of potentially relevant news events from the KnowledgeStore, based on the user query.
- **Event Filtering**: uses a classifier to filter the events found in the search step based on their expected usability (i.e., keeps only well-formed events that are suited for further processing). Uses features defined on the user query and information from the KnowledgeStore.
- **Event Ranking**: ranks the events according to their expected relevance, based on both the user query and the user model. Uses the user query, the user model, and information from the KnowledgeStore to define features for the ranking regressor. Selects the event with the highest predicted relevance value.
- **Summary Creation**: creates a summary sentence for the selected event by extracting the sentence in which the selected event was mentioned.

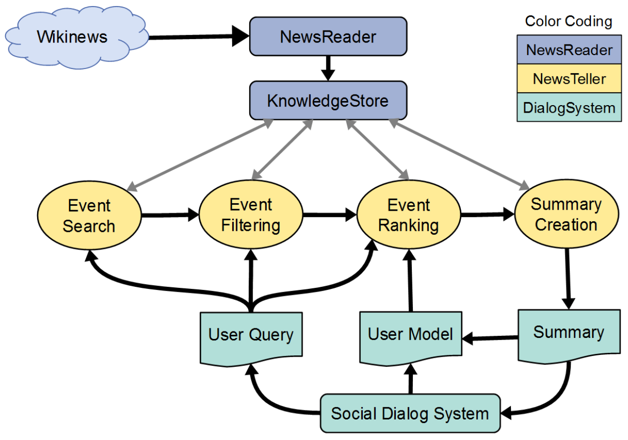

Figure 1: Overall architecture of the NewsTeller system.

Both the first and the last step of this pipeline have a straightforward implementation and will not be discussed in further detail in this paper. Instead, we will focus on the filtering and the ranking step. Both of them were implemented by using machine learning techniques and will be described in the following subsections.

3.2 Event Filtering

Just like any NLP system, the NewsReader NLP pipeline is not perfect. This means that not all events extracted by the pipeline and stored in the KnowledgeStore are suitable for further processing. For example, some actors might be missing (e.g., an event ⟨*Barack Obama – talk to*⟩ missing the object of the original sentence "Barack Obama talked to Matteo Renzi"), or two semantically distinct events might accidentally be merged into one. Therefore, a subsequent filtering step is necessary to eliminate events of low quality. This filtering is performed by a classifier.

As training data set for the filtering problem, about 6,000 events labeled with the two classes USABLE and NOT USABLE were used. This data set contains only about 16.01% of USABLE events. All events belonging to the NOT USABLE class were further annotated with respect to the reasons why they were considered to be NOT USABLE. We identified 10 different reasons, five of syntactic nature (e.g., missing an object) and five of semantic nature (e.g., merged events). Features were engineered looking at each of these reasons for non-usability individually. For instance, one feature for the "missing object" reason compares the number of actors associated with the event to the number of expected arguments for the event's verb.

We compared two different classifiers: an ensemble of specialized classifiers (one for each reason of non-usability) and a global random forest. For each of the specialized classifiers, one to six features were selected based on a combination of different feature selection algorithms (e.g., RELIEF-F, gain ratio, and wrappers). Each individual classifier was trained on the union of all USABLE events and the NOT USABLE events marked with the respective reason. The logical "AND" is used as the ensemble's aggregation rule, i.e., the ensemble only outputs USABLE if all of its members output USABLE. The global random forest was trained on the overall data set using the union of the specialized classifiers' feature sets (28 features in total).

Classifier	BA	κ	F_1
Random 50:50	0.5000	0.0000	0.2425
Always No	0.5000	0.0000	NaN
Ensemble	0.7497	0.4764	0.5649
Random Forest	0.7645	0.5411	0.6125

Table 1: Table showing the overall classifier performance on the USABLE – NOT USABLE problem for the different approaches. The columns show the metrics Balanced Accuracy, Cohen's κ, and F_1 score.

Note that due to the small fraction of USABLE events, there is a considerable class imbalance in the given data set. We used cost-sensitive classification to counteract this class imbalance by defining a higher penalty for false negative classification errors than for false positive errors. The classifiers were then trained with the objective to minimize the overall cost instead of maximizing the classification accuracy. The respective cost matrices were hand-tuned after feature engineering.

Both approaches were evaluated on the data set using a 10-fold cross-validation. We use the following metrics which are considered to be fairly robust against class imbalance. They are based on the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), as well as on the accuracy metric:

- **F_1 measure**: $F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$
  The F_1 measure (also called F_1 score) is the harmonic mean of $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$.
- **Balanced accuracy**: $BA = \frac{1}{2} \cdot \frac{TP}{TP+FN} + \frac{1}{2} \cdot \frac{TN}{TN+FP}$
  The balanced accuracy is the unweighted average of the recall for the positive class and the recall for the negative class.
- **Cohen's kappa**: $\kappa = \frac{p_0 - p_e}{1 - p_e}$
  Cohen's kappa adjusts the classifier's accuracy p_0 by taking into account the probability p_e of random agreement between the classifier and the ground truth.

Table 1 shows the results of the 10-fold cross-validation for a 50:50 baseline, an "always no" baseline, and the two classifiers. As one can see, both classifiers perform considerably better than the baselines.

It is interesting to observe that the random forest classifier seems to have a better performance than the ensemble-based approach. This might be at least partially due to the following differences: Each specialized classifier from the ensemble uses both a smaller amount of training data and a smaller feature set than the global random forest classifier. Moreover, due to the use of the logical "AND" as aggregation function, classification errors in the ensemble might accumulate.

As the random forest classifier showed considerably better performance than the ensemble-based approach, it was chosen for the final system.

3.3 Event Ranking

After having filtered the events, they need to be ranked according to their predicted relevance. We used the "learning to rank" approach that was introduced in Section 2. A data set of about 3,100 events was created. 2,000 of them were labeled by the first author of this paper in an objective way (i.e., without taking into account personal interests), the remaining events were labeled by 11 different annotators, each of them having different general interests. Each annotator created a set of queries to the system and labeled a set of example events for each of these queries. We used four ordered classes: IRRELEVANT (index 0), PARTIALLY RELEVANT (index 1), RELEVANT (index 2), and VERY RELEVANT (index 3). Figure 2 illustrates the label distribution of this data set.

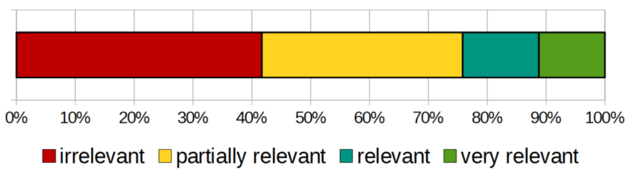

Figure 2: Label distribution in the ranking data set.

There were three groups of features that were used in this regression problem:

- **Event Features**: features based only on the event representation itself, e.g., the length of the sentence from which the event was extracted.
- **Query-Event Features**: features comparing the user query to the event description, e.g., by using word embeddings.
- **Interest-Event Features**: features comparing the user's general interests to the event description, again e.g., by using word embeddings.

As each event was labeled by only one annotator, we could not define features based on ratings of other users (as it is done in "collaborative filtering" recommender systems).

We used a random forest regressor because research indicates that they are well-suited for the "learning to rank" problem [10]. Two different regressors were trained: One of them was only allowed to use features from the first two feature groups. It is called the NOUM regressor (short for "**NO** **U**ser **M**odel"). The other regressor was allowed to use features from all three feature groups and is called the UM regressor (short for "**U**ser **M**odel"). By comparing the performance of these two regressors, we expect to find some indication about whether taking user interests into account can help to solve the event ranking problem.

A leave-one-out procedure on the annotator level was used to evaluate both ranking-based approaches on the ranking data set: We iterated over all different annotators, using all events labeled by the respective annotator as test set and the remaining events as training set. Results were averaged across the iterations.

For evaluating "learning to rank" approaches, one can for instance use the $Precision@k$ metric, which is defined as follows [8]:

$$Precision@k = \frac{\#\{\text{relevant documents in top k positions}\}}{k}$$

Based on this general ranking metric, we defined two application specific metrics. Let Q be the set of queries. For each query $q \in Q$, let $\alpha(q)$ denote the ground truth label of the event that was put on top of the list by the regressor, i.e. the event with the highest predicted relevance.

Regressor	$P_{>0}$	$P_{>1}$
RANDOM	0.5842	0.2079
NOUM	0.7663	0.4238
UM	0.7723	0.4842

Table 2: Table showing the results of evaluating the two regression-based approaches on the ranking data set.

Regressor	$P_{>0}$	$P_{>1}$
RANDOM	0.6042	0.2250
NOUM	0.7625	0.2667
UM	0.7375	0.2708

Table 3: Table showing the results of evaluating the three ranking approaches in the user study.

Let further $|A|$ denote the cardinality of set A. Then, our metrics can be defined as follows:

- **Precision greater than zero:** $P_{>0} = \frac{|\{q \in Q : \alpha(q) > 0\}|}{|Q|}$
  This metric measures the frequency with which the top element of the list is at least PARTIALLY RELEVANT. It corresponds to the question how successful the regressor is in avoiding IRRELEVANT events.
- **Precision greater than one:** $P_{>1} = \frac{|\{q \in Q : \alpha(q) > 1\}|}{|Q|}$
  This metric measures the frequency with which the top element of the list is at least RELEVANT. It corresponds to the question how often the event selected by the regressor is a "good" response to the user query.

Table 2 compares the results of an annotator-level leave-one-out evaluation for a random baseline (which simply shuffles the events) to the results of the two regression-based approaches. As one can see, both regression-based approaches perform better than the baseline. Although the UM regressor outperforms the NOUM regressor with respect to the $P_{>1}$ metric, there is hardly any difference between the two regressors with respect to the $P_{>0}$ metric.

We interpret these results in the following way: A random forest regressor is capable of solving the ranking problem to a sufficient degree as it considerably outperforms a simple baseline. Using interest-based features does not affect the regressor's capability of avoiding IRRELEVANT events, but it seems to improve the detection of RELEVANT and VERY RELEVANT events.

4 User Study

In order to validate the NewsTeller system in an online scenario, we conducted a user study with 48 participants (29 male, 19 female; on average 29.02 years old). The user study was conducted as an online survey. Participants were asked to describe their general interests and to formulate five queries to the NewsTeller system. For each of their queries, they received responses from three different system configurations, based on the RANDOM baseline and the regression approaches NOUM and UM. Participants did not know which configuration generated which response. They were asked to label each of the responses with respect to the relevance classes used in Section 3.3.

Table 3 shows the $P_{>0}$ and $P_{>1}$ metrics for the three system configurations computed on the labels collected in the user study.

With respect to the $P_{>0}$ metric, one can observe similar effects as on the ranking data set: Both regression-based approaches considerably outperform the baseline. A χ^2 significance test (with a significance level of $\alpha = 0.05$) showed that these differences are statistically significant. The small difference between the two regression-based approaches is however not statistically significant. This confirms the results from Section 3.3 with respect to the $P_{>0}$ metric: both regressors are better than the baseline and yield comparable performance.

Looking at the $P_{>1}$ metric, one can observe only relatively small differences between the three approaches. We observed no statistically significant differences between any of the approaches. There seems to be only a weak tendency for both the NOUM as well as the UM regressor to outperform the RANDOM baseline. The small difference between the two regression-based approaches seems to be mainly caused by random noise.

These results somewhat contradict what we observed in Section 3.3 with respect to the $P_{>1}$ metric: On the ranking data set, both regression-based approaches considerably outperform the baseline, and the UM regressor yields a better performance than the NOUM regressor. In the user study, however, we cannot find any significant differences between the three approaches, and their respective values are very similar.

We suppose that the lack of statistical evidence with respect to the $P_{>1}$ metric can be attributed to overfitting: The regressors were trained on a relatively small data set of about 3,100 data points. This data set might be simply too small to generalize properly to unseen data. Moreover, the set of annotators that generated the ranking data set might not accurately reflect the overall population of potential users, e.g. by being too homogeneous. That way, the results obtained on the data set might be too optimistic.

Although we were able to confirm that the regression-based systems are more successful in avoiding IRRELEVANT events than a random baseline (indicated by the $P_{>0}$ metric), we could not observe any differences with respect to the detection of RELEVANT and VERY RELEVANT events. We were thus also not able to confirm that the use of interest-based features can help to improve the detection of RELEVANT and VERY RELEVANT events. This urges for further research.

5 Conclusion

In this paper, we presented the NewsTeller system, a personalized retrieval system for news events that can be used to initiate news-related small talk in a social dialog system. By using relatively simple approaches, we were already able to achieve reasonable performance. Although the results of a user study partially contradict what was observed on a data set, we think that our results are promising and that further research in this direction is worthwhile.

Some obvious starting points for future work are the improvement of the regressors' generalization capability (e.g., by training them on larger data sets), the actual integration of the NewsTeller system into the social dialog system presented in Section 2, and the extension of the system's scope to multiple dialog turns by allowing follow-up questions about previously presented news events.

Acknowledgments

This paper is based on a Master's thesis [17] which was written in cooperation of the Interactive Systems Lab (ISL) at the Karlsruhe Institute of Technology (KIT) and the Human Language Technology (HLT) research group at Fondazione Bruno Kessler (FBK). We would like to thank Marco Rospocher and Anne-Lyse Minard from FBK for their valuable input to this project.

References

[1] B. Reeves and C. Nass, *How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge university press, 1996.

[2] M. Schmidt, J. Niehues, and A. Waibel, "Towards an open-domain social dialog system," in *Proceedings of the Seventh International Workshop on Spoken Dialogue System*, 2016.

[3] M. F. McTear, "Spoken dialogue technology: enabling the conversational user interface," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 90–169, 2002.

[4] J. H. Martin and D. Jurafsky, *Speech and language processing*, ch. Dialogue and Conversational Agents. Prentice Hall, 2nd ed., 2008.

[5] R. Agerri, E. Agirre, I. Aldabe, B. Altuna, Z. Beloki, E. Laparra, M. L. de Lacalle, G. Rigau, A. Soroa, and R. Urizar, "Newsreader project," in *30th Conference of the Spanish Society for Natural Language Processing (SEPLN)*, 2014.

[6] P. Vossen, G. Rigau, L. Serafini, P. Stouten, F. Irving, and W. R. V. Hage, "Newsreader: recording history from daily news streams," in *Proceedings of the 9th Language Resources and Evaluation Conference (LREC2014)*, (Reykjavik, Iceland), May 26-31 2014.

[7] F. Corcoglioniti, M. Rospocher, R. Cattoni, B. Magnini, and L. Serafini, "Interlinking unstructured and structured knowledge in an integrated framework," in *7th IEEE International Conference on Semantic Computing (ICSC), Irvine, CA, USA*, 2013.

[8] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[9] L. Hang, "A short introduction to learning to rank," *IEICE TRANSACTIONS on Information and Systems*, vol. 94, no. 10, pp. 1854–1862, 2011.

[10] A. Mohan, Z. Chen, and K. Q. Weinberger, "Web-search ranking with initialized gradient boosted regression trees.," in *Yahoo! Learning to Rank Challenge*, pp. 77–89, Citeseer, 2011.

[11] K. Yoshino, S. Mori, and T. Kawahara, "Spoken dialogue system based on information extraction using similarity of predicate argument structures," in *Proceedings of the SIGDIAL 2011 Conference*, pp. 59–66, Association for Computational Linguistics, 2011.

[12] K. Yoshino and T. Kawahara, "Information navigation system based on pomdp that tracks user focus," in *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 32, 2014.

[13] K. Yoshino and T. Kawahara, "News navigation system based on proactive dialogue strategy," in *International Workshop series on Spoken Dialogue Systems technology*, 2015.

[14] A. Díaz and P. Gervás, "User-model based personalized summarization," *Information Processing & Management*, vol. 43, no. 6, pp. 1715–1734, 2007.

[15] A. Díaz, P. Gervás, A. García, and L. Plaza, "Development and use of an evaluation collection for personalisation of digital newspapers," in *LREC*, 2010.

[16] D. Billsus and M. J. Pazzani, "User modeling for adaptive news access," *User modeling and user-adapted interaction*, vol. 10, no. 2-3, pp. 147–180, 2000.

[17] L. Bechberger, "Personalized news event retrieval for small talk in social dialog systems," Master's thesis, Karlsruhe Institute of Technology (KIT), 2016.