Towards Context-Aware Personalization and a Broad Perspective on the Semantics of News Articles

Jeremy Jancsary
Austrian Research Institute
for Artificial Intelligence
Freyung 6/6
A-1010 Vienna, Austria
jeremy.jancsary@ofai.at

Friedrich Neubarth
Austrian Research Institute
for Artificial Intelligence
Freyung 6/6
A-1010 Vienna, Austria
friedrich.neubarth@ofai.at

Harald Trost
Section for Artificial Intelligence
Medical University of Vienna
Freyung 6/2
A-1010 Vienna, Austria
harald.trost@meduniwien.ac.at

ABSTRACT

We analyze preferences and the reading flow of users of a popular Austrian online newspaper. Unlike traditional news filtering approaches, we postulate that a user's preference for particular articles depends not only on the topic and on propositional contents, but also on the user's current context and on more subtle attributes. Our assumption is motivated by the observation that many people read newspapers because they actually *enjoy* the process. Such sentiments depend on a complex variety of factors. The present study is part of an ongoing effort to bring more advanced personalization to online media. Towards this end, we present a systematic evaluation of the merit of contextual and non-propositional features based on real-life clickstream and postings data. Furthermore, we assess the impact of different recommendation strategies on the learning performance of our system.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*text analysis*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*user profiles and alert services, performance evaluation (efficiency and effectiveness)*

General Terms

Algorithms, Human Factors, Languages

1. INTRODUCTION

Why readers of online newspapers prefer some articles over others is not understood very well so far. Obviously, content plays an important role, but as we know from studies concerned with printed media [10], it only accounts for about 40% of a story's satisfaction rating. According to [10, 9], other factors can be as diverse as readability concerns, writing style, the type of a story, visual complexity, or proper use of photography. Online media differ from printed media

in various respects. Studies suggest that online users have limited patience for locating information in deeply nested website hierarchies [8]. Indeed, web logs show that the majority of requests are for news stories on the front page.

In the MAGNIFICENT project, we aim at a better understanding and modeling of the reading preferences of individual users or user groups in the setting of the Austrian online newspaper `derStandard.at`. The goal is to gain deeper insights into both the relevant parameters of stories and the adaptive training of user profiles along these parameters. Ultimately, we want to improve a user's reading experience through personalized presentation of articles, taking into account personal reading behavior and use of the medium.

Most modern news filtering systems either employ collaborative filtering strategies, or they focus on the contents of articles. Often, systems in the latter category assume that the topic of an article is the primary if not the only factor that determines user satisfaction. This view is too narrow; other parameters contribute significantly to a user's preference for certain news articles, in particular if reading an online newspaper is seen in part as a recreational activity rather than professional acquisition of information.

Our approach is novel in that we aim at gaining results about the preference of users, rather than topical relevance of certain articles. From a broader perspective, we emphasize a comprehensive view of semantics. The meaning of text goes far beyond the basic propositional content. It comprises a rich mix of factors, including rhetorical structure, style, standpoint, and many other aspects writers routinely convey to their readers.

Moreover, contextual information has to be considered if we are to recommend a sequence of articles that is in accordance with a user's preference [1]. Such contextual information exists in the form of previously read articles in a user's current session, for instance. In fact, the availability of data about reading behavior, and the potential to provide for personalized presentation, are major advantages of online newspapers over their printed counterparts. The present paper describes a first study in our journey towards better understanding of reader preference.

2. APPROACH

We maintain a separate preference model for each user. The model itself is in essence a discriminative binary classifier that is updated in an online fashion and draws on several features. Recommendations are then computed by classifying eligible news articles in real-time and inspecting their

confidence scores. This real-time classification allows us to incorporate contextual features such as the current time or information about the previously read article. The model is updated whenever we receive feedback about the quality of the previous recommendation; such feedback can be obtained in an implicit fashion or through explicit ratings.

While the above approach is very general, it mandates a number of choices, most notably with regard to classification algorithms, features, and recommendation strategies. We will discuss our choices presently.

2.1 Recommendation Setting

For our experimental study, we consider a total of T recommendations. At each round t, a new article must be suggested, where our system can initially choose from a pool of T positive examples and T negative examples (this classification is of course unknown to the system beforehand). After suggesting an article, its "true" class will be revealed such that the system can adapt its model.

We then measure the cumulated regret of our system over the whole sequence with respect to a hypothetic system that suggests only positive examples. Towards that end, each negative example suggested by our system will be counted as a mistake. The regret incurred for a particular mistake need not be uniform over time. In our experiments, we consider uniform weighting and the following schemes:

$$\omega_a(t) = \frac{1}{2} + Q\left(-\frac{2C}{T}t + C\right),$$
$$\omega_d(t) = \frac{1}{2} + Q\left(\frac{2C}{T}t - C\right),$$

where $Q(x) = 1 - \Phi(x)$ is the tail probability of the standard normal distribution and $C \triangleq 3$ is a constant controlling the slope. These functions model ascending regret and descending regret per mistake and are depicted next to tables 1 and 2, respectively. Our weighting schemes preserve the property that the expected average regret of a system that chooses examples at random will be 0.5. Although somewhat arbitrary, the intention behind our weighting choices is to cover the fact that some users are more sensitive to mistakes in the beginning, whereas others rate them more severely after some time, when they expect the system to have captured their preferences. Even though our notion of average regret corresponds to an error rate in the case of uniform weighting, it should in general *not* be compared to offline classification error rates.

2.2 Online Learning Algorithms

Discriminative online learning algorithms have a long history, with the Perceptron [11] reaching back as far as 1958. Most of these algorithms work by updating a linear model in rounds. At each round, a new example and its class (positive or negative) becomes available. The algorithm can then choose to update the model parameters, based on its current capability to accurately predict the class of the new example. The family of Passive-Aggressive (PA) algorithms [2] is a recently introduced variant that has proved particularly successful. PA updates ensure that an example is correctly classified at least by a certain margin. In order to deal with data that is not linearly separable, kernel methods have also been applied to online learning. A practical member of this family is the Forgetron [4], a variant of the kernel-based Perceptron which operates on a fixed memory budget.
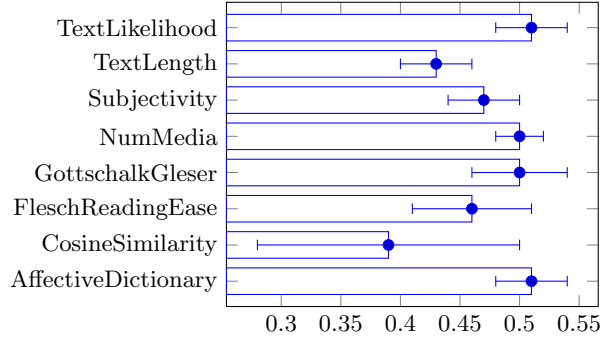

Figure 1: Regret incurred by some individual features using a Perceptron—lower is better.

2.3 Feature Engineering

We implemented feature extraction modules for a variety of facets in order to asses their impact. The features were combined in a linear or kernel-based model, depending on the online learning algorithm in use.

2.3.1 Non-propositional features

In particular, we were eager to find features that capture aspects of user preference that go beyond the mere topic of an article, which we hypothesized might add some additional value over a pure vector-space approach.

AffectiveDictionary: We implemented a scoring routine based on the "Affective Dictionary Ulm" [7], thereby assigning scores related to affective aspects such as anger, contentedness, fear, etc. to articles.

GottschalkGleser: For each article, we computed its Gottschalk-Gleser scores [6]. These are related to the above, although based on a different dictionary.

TextLikelihood: Based on an existing unigram language model, we computed the log-likelihood of the article text, which served as a proxy measure for complexity of words.

FleaschReadingEase: The idea here was to measure reading ease of articles using the Flesch-Kincaid score [5].

TextLength: We measured the length of articles as their number of tokens.

Subjectivity: Beforehand, we trained a simple classifier on a hand-selected set of articles which was split into "subjective" and "non-subjective" articles. This classifier was used to assign subjectivity scores to new articles.

NumMedia: The number of media files in an article.

2.3.2 Context-sensitive features

In addition, we defined the following features which exploit the current context of the user, or the current state of the model, and must thus be computed in real-time.

CurrentTime: Binary features that encode the day of the week and a real value encoding the current hour.

CategoryFlow: Binary features capturing the transition from the category of the previous article to that of the candidate article.

PreferenceFlow: Unigram, bigram and trigram features that encode whether the user liked or disliked the previous one, two or three articles suggested by the system.

Table 1: Postings experiment at ascending mistake weighting (numbers indicate regret)

$\omega_a(t)$ curve (ascending weighting).
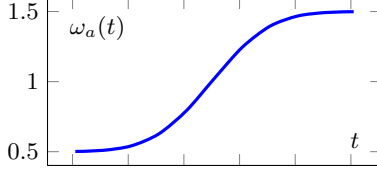
	Top-Rated		Boundary	
	PA-α	PA-γ	PA-α	PA-γ
Cos	**0.38 ± 0.10**	0.39 ± 0.10	0.46 ± 0.03	0.47 ± 0.02
NonProp	0.47 ± 0.03	0.48 ± 0.03	0.40 ± 0.06	**0.39 ± 0.06**
All	0.47 ± 0.02	0.46 ± 0.02	0.40 ± 0.06	**0.39 ± 0.06**

Table 2: Postings experiment at descending mistake weighting (numbers indicate regret)

$\omega_d(t)$ curve (descending weighting).
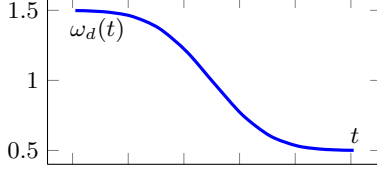
	Top-Rated		Boundary	
	PA-α	PA-γ	PA-α	PA-γ
Cos	**0.36 ± 0.13**	0.36 ± 0.14	0.47 ± 0.02	0.48 ± 0.02
NonProp	0.45 ± 0.06	0.45 ± 0.05	0.41 ± 0.05	**0.41 ± 0.04**
All	0.44 ± 0.04	0.45 ± 0.04	0.42 ± 0.05	**0.42 ± 0.04**

ContentFlow: Cosine similarity computed between a TF-IDF-weighted vector-space representation of the previously read article and the candidate article.

CosineSimilarity: Cosine similarity between a TF-IDF vector-space representation of the candidate article and a regularly adapted TF-IDF preference model of the user.

2.4 Recommendation Strategies

Successful recommendation involves optimization of two conflicting objectives: On the one hand, we would like to carefully avoid erroneous recommendations; on the other hand, we would like for our system to learn as rapidly as possible, which requires exploration of articles the system is unsure about. The most successful recommendation strategy certainly depends on the weighting of mistakes over time. In this paper, we followed two strategies.

2.4.1 Top-rated articles

At each round, we chose a sample of at most 100 eligible articles and recommended the one article which received the highest confidence rating by the classifier. This strategy is rather conservative and tries to avoid mistakes at all cost.

2.4.2 Articles close to the decision boundary

We also experimented with a more aggressive strategy. Again, we sampled at most 100 eligible articles and picked the article which was closest to the decision boundary, on the positive side. Similar strategies have been used successfully in active learning to minimize the number of examples required for a good hypothesis [3]. Compared to the above strategy, this promotes fast learning, but at a higher risk.

3. EXPERIMENTS

Since our project is still in the development phase, explicit preference ratings by users were not available, yet. Hence, we defined surrogate measures for preference which we expected to be correlated and which could be extracted readily from a substantial amount of anonymized real-life data collected by the online newspaper.

3.1 Postings data

Most articles of the online newspaper allow for posting of user comments. The hypothesis underlying our surrogate measure for preference was then that users post comments under articles that are of interest to them or capture their attention. For each heavy user of the site, we thus built a substantial balanced dataset consisting of positive articles (i.e., articles with an associated posting by the user) and an equal number of negative articles. For each positive article, we obtained a negative counterpart by choosing an article from the same day that attracted a large number of postings by *other* users. The size of the datasets varied between roughly 3,400 and 4,700 labeled articles for each user. Using these datasets, we went about simulating the recommendation scenario outlined in section 2.1. The time and sequence of suggestions were not actually controlled by the users in this setting, hence many context-sensitive features described in section 2.3.2 were unapplicable in this experiment.

First, we wanted to gain a rough idea of the discriminative power of some individual features. We used the Perceptron algorithm and top-rated recommendation; the results are shown in figure 1. The bars indicate the average regret incurred using uniform mistake weighting. While some tendencies regarding the features are already visible in these first results, we expected recommendation strategies, regret weighting and learning algorithms to have a major impact.

We proceeded with a more systematic evaluation and established three major feature sets: Cos, which is simply the CosineSimilarity feature described above, NonProp, which includes all non-propositional features of section 2.3.1, and All, which combines the two aforementioned sets. Cos corresponds more or less to a traditional content-based approach, and we wanted to measure how a combination of our non-propositional features fares in comparison.

Moreover, we combined the two different recommendation strategies outlined in section 2.4 with Passive-Aggressive learning at two different parameterizations, which we refer to as PA-α and PA-γ. The first corresponds to PA-II updates using $C = 100$, while the latter uses $C = 0.01$. Meta parameter C essentially impacts the aggressiveness of the updates; PA-α corresponds to a rather aggressive update scheme whereas PA-γ applies more conservative updates.

Finally, for each of the above combinations of feature sets, classification algorithms and recommendation strategies, we weighted the mistakes using the ascending and descending schemes described in section 2.1. Tables 1 and 2 show the average regret achieved by the competing systems, averaged over 7 users, along with the standard deviation over users. The weighting curves are depicted next to the tables.

From the results, we observe the following main effects: (a) irrespective of regret weighting, the Cos feature consistently yields the single best system—but its effectiveness can vary strongly over users compared to NonProp and All, which are competitive in several systems; (b) the difference between NonProp and All is negligible in all cases, so there is no gain in combining the non-propositional features with the traditional cosine similarity measure; (c) close-to-boundary recommendation works better when regret is ascending, which confirms that this strategy eventually leads to faster learning; (d) similarly, top-ranked recommendation works well when the cost of mistakes is high initially and then decays; (e) NonProp works considerably better when combined with close-to-boundary recommendation whereas top-rated recommendation is more appropriate for Cos; (e) PA is relatively robust regarding the choice of C.

We also obtained numbers for all systems under uniform regret weighting, but the results are consistent with the above discussion so we omit them here due to a lack of space.

3.2 Clickstream data

We next considered clickstream data obtained from the online newspaper. Here, the setup deviated slightly from the recommendation setting described in section 2.1 in that the articles chosen by the user were already determined through the clickstream log so that we did not actually have to recommend articles. Instead, we chose *page viewing time* as a surrogate measure for preference and tried to predict whether a user would stay on a given article for longer than his median page viewing time. The number of positive and negative examples was thus balanced by construction. As opposed to the previous setting, all contextual features were applicable here. Moreover, we extended our set of classification algorithms by the Perceptron and the kernel-based Forgetron algorithm with a parameter-free RBF kernel of the form $\exp(-\frac{1}{2}||x_a - x_b||^2)$.

Table 3 shows the results of the experiment for several feature combinations, averaged over 47 users. While we cannot describe all feature sets in detail, the take-home points are: (a) disappointingly, the system drawing on a single bias feature that is always 1.0 obtains the best score; (b) we observe in the data that some users seem to have "phases" where they simply click through articles in a haste, irrespective of the article contents; (c) the best way of predicting the class of the current article is thus to simply predict the class of the previous article, which is exactly what is achieved by "flipping" the weight for the bias term; (d) the page viewing time does not appear to be an appropriate surrogate measure for user preference. Moreover, PA algorithms obtain the best scores, while the more expensive Forgetron does not help.

4. CONCLUSION

We assessed novel non-propositional features and showed that they are competitive with a traditional content-based approach in several cases, while cheaper to compute. The best recommendation strategy seems to depend on the set of features in use, as well as the weighting of mistakes over time. Moreover, we established that PA classification algorithms are a good choice in our scenario. While a combination of non-propositional and content-based features does not show substantial gains, one should keep in mind that we only used surrogate measures in these experiments; their actual correlation with true preference is so far unknown.

Table 3: Results of the clickstream experiment

	Perceptron	Forgetron	PA-α	PA-γ
Flow	0.46 ± 0.03	0.47 ± 0.03	0.47 ± 0.03	$\mathbf{0.42} \pm 0.06$
Cos	0.43 ± 0.05	0.43 ± 0.05	$\mathbf{0.42} \pm 0.05$	0.49 ± 0.03
Ctx	0.46 ± 0.04	0.46 ± 0.04	0.46 ± 0.04	$\mathbf{0.42} \pm 0.05$
Bias	0.43 ± 0.05	0.43 ± 0.05	0.42 ± 0.05	0.46 ± 0.03
Invar	0.43 ± 0.04	0.46 ± 0.03	$\mathbf{0.43} \pm 0.05$	0.44 ± 0.05
All	0.46 ± 0.04	0.46 ± 0.03	0.46 ± 0.04	$\mathbf{0.43} \pm 0.05$

Recently, we have begun deploying our system on the live website of `derStandard.at`; we will now collect explicit preference ratings and implicit feedback of selected users on an opt-in basis. It will be interesting to analyze the impact of non-propositional features and different recommendation strategies in this more natural setting.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE*, 17:734–749, 2005.

[2] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *JMLR*, 7:551–585, 2006.

[3] S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. *JMLR*, 10:281–299, 2009.

[4] O. Dekel, S. Shalev-Shwartz, and Y. Singer. The Forgetron: A kernel-based perceptron on a budget. *SIAM Journal on Computing*, 37:1342–1372, 2008.

[5] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

[6] L. A. Gottschalk and G. C. Gleser. *The measurement of psychological states through the content analysis of verbal behavior*. University of California, 1969.

[7] M. Hölzer, N. Scheytt, and H. Kächele. Das "Affektive Diktionär Ulm" als eine Methode der quantitativen Vokabularbestimmung. In *Textanalyse – Anwendungen der computerunterstützten Inhaltsanalyse*, pages 185–212. Westdeutscher Verlag, 1992.

[8] J. Palmer. Designing for web site usability, 2002.

[9] Readership Institute. Newspaper content: What makes readers more satisfied. `http://www.readership.org/content/editorial/data/what_content_satisfies_readers.pdf`, 2001.

[10] Readership Institute. Inside satisfaction: What it means, how to increase it. `http://www.readership.org/content/editorial/data/elements_of_satisfaction.pdf`, 2002.

[11] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.