

CrowdTargeting: Making Crowds More Personal

Joana Costa, Catarina Silva IEEE member, Bernardete Ribeiro IEEE member, Mário Antunes

Abstract—Crowdsourcing is a bubbling research topic that has the potential to be applied in numerous online and social scenarios. It consists on obtaining services or information by soliciting contributions from a large group of people. However, the question of defining the appropriate scope of a crowd to tackle each scenario is still open. In this work we compare two approaches to define the scope of a crowd in a classification problem, casted as a recommendation system. We propose a similarity measure to determine the closeness of a specific user to each crowd contributor and hence to define the appropriate crowd scope. We compare different levels of customization using crowd-based information, allowing non-experts classification by crowds to be tuned to substitute the user profile definition. Results on a real recommendation data set show the potential of making crowds more personal, i.e. of tuning the crowd to the crowdtarget.

Keywords-Crowdsourcing, Recommendation Systems, Customization, Text Classification

I. INTRODUCTION

Recommendation systems have become very popular in a wide range of Internet applications. The general idea consists on giving a recommendation to each user about a particular product, according to its preferences. For example, in a book recommendation system, a user may be influenced to buy a given book based on the comments produced by individuals that share common interests or the same cultural and intellectual background.

Model customization for a recommendation system in such a distributed environment like Internet, can be very complex and challenging. The contextual environment in which the recommendation is made changes through time and the model has to adjust to those variations accordingly. Also, the heterogeneity and diversity of Internet users, each one with a different *profile*, challenge even more the model, as it has to tackle simultaneously with distinct *groups* of users and drifting contexts. From a research point of view two important and emergent lines of research in machine learning

and data mining can be put forward: (i) tracking user's preference through time and (ii) customizing models produced to fulfil user information needs based on the adjusted personal preferences. [1].

Profiling [2] is one of the most simple and common techniques used to customize a user model. With this technique, user's preferences are defined according to the previously acquired information related to the contextual groups each user belongs to. The major drawback on profiling is the difficulty to define rigorously the boundaries to more subjective problems, like those involving personal preferences and emotional issues.

An emerging and challenging solution consists on taking advantage of *crowdsourcing* [3], [4]. It consists on a distributed classification method in which a crowdsourcer submits a complex task to groups of people, termed *crowds*, in order to obtain different solutions for further analysis and evaluation. The core idea of using crowdsourcing is to learn from previously seen contexts and to use the low cost workforce of the users that already gave their feedback to the recommendation system in order to deal with the newly seen users. The crowdsourcing paradigm has been enabled by Web technologies and its applications use a distributed computer infrastructure and cloud computing facilities provided by the Internet [5].

Crowdsourcing has the following main interesting features that may benefit the user's preferences adjustment in a dynamic recommendation system: the fact that it is a complex and widely distributed classification infrastructure; the implicit heterogeneity provided by the crowd members and the intrinsic ability to cope with dynamic variations of context through time in a distributed and heterogeneous way, in a particular application domain. Moreover, this *non-expert knowledge* provided by a heterogeneous crowdsourcing scenario, which produces distinct and unrelated examples, can therefore be a source of valuable input to learning systems based on more traditional machine learning methods, e.g. kernel-based methods [6], such as Support Vector Machines (SVM) [7], [8]

In dynamic recommendation systems there is usually not enough labeled data. To tackle this issue active learning methods allow us to design learning algorithms that may effectively filter or choose a subset of examples for being further labeled by a supervisor (*oracle*). By using this strategy we are able to expedite the learning process and thus to reduce the labeling efforts required by the supervisor [9]. At the

Joana Costa, Catarina Silva and Mário Antunes are with the School of Technology and Management, Polytechnic Institute of Leiria, Portugal ({joana.costa,catarina, mario.antunes}@ipleiria.pt).

Catarina Silva and Bernardete Ribeiro are with Department of Informatics Engineering, CISUC, University of Coimbra, Portugal ({catarina,bribeiro}@dei.uc.pt).

Mário Antunes is with the Center for Research in Advanced Computing Systems (CRACS), University of Porto, Portugal (mantunes@dcc.fc.up.pt).

same time, and more appropriate to adjust users' preferences, active learning allow each user to define personal labels and then to build a customized learning model that better fits his preferences.

In this work we propose a framework to deal with customization in recommendation systems using crowd-based non-experts. The general framework aims to use an active learning strategy based on both customized and personal models, in order to improve over an SVM base line model previously proposed and validated by the authors in [10], [11], [12]. The proposed framework allows non-experts classification performed by crowds to be tuned, for further user profile adjustment, thus mitigating the labeling effort normally requested to the user. A case study classification scenario is used to test and validate our efforts, even though the framework is designed to be generic and applicable to different application domains. The main analysis to carry out in this research consists on determining if tuned crowds in an active learning approach are suitable to retrieve customized user preferences.

The rest of the paper is organized as follows. In Section II we introduce the necessary background to our work, namely the definition of crowdsourcing, including the discussion of its applicability in learning systems and the discussion of social networks as crowd-based data sources. Section III presents the proposed framework, followed by experimental setup and results in Sections IV and V respectively. Finally, Section VI presents some conclusions and future work.

II. BACKGROUND

In this section we present and describe the main background knowledge that correspond to the foundations upon which our work has been grounded. We present the generic background on crowdsourcing for a better understanding of the approach proposed in this paper. We further discuss the appropriateness of using crowds as a source of non-expert knowledge in dynamic learning systems.

A. Crowdsourcing

Surowiecki's seminal work [4] introduces the concept of *crowdsourcing*. The underpinning idea is that, under the right circumstances, groups can be remarkably intelligent and efficient and are often smarter than the smartest individual in them. That is, their decisions are usually better than the decisions of the brightest party in the group [4], [13].

Besides the intelligent use of a group, another noteworthy advantage on using crowdsourcing is the notorious difficulty for an algorithm to perform some classification tasks that are quite simple and intuitive for humans, like speech or image recognition, language understanding, text summarization and

labeling [14]. Taking advantage of these inherent capabilities, many crowdsourcing platforms emerged, such as the now widely used Amazon Mechanical Turk and Yahoo! Answers. In the next section, we will address some known applications of crowdsourcing based on these approaches.

B. Active non-expert learning

Active learning strategies employ machine learning algorithms to achieve greater accuracy with fewer training label samples. That is, an active learner may query data instances to be labeled by the supervisor and then to choose data from the learning model [15].

The reason for using active learning is mainly to expedite the learning process and thus to reduce the labeling efforts required by the supervisor. An active learning strategy has been successfully applied in many modern machine learning problems where data is abundant but unbalanced, and labels are usually scarce or expensive to obtain [9], [15].

Depending on the selection strategy, active learning methods can be defined as committee-based or certainty-based [16]. In committee-based methods, the active examples are the combination of the individual classification obtained by each committee member, by determining those in which the members disagree the most as the candidates to be labeled [17]. Certainty-based methods try to determine the most uncertain examples according to the learning method used, as being the active examples to be labeled.

Crowdsourcing and active learning strategies can be successfully put together in the same learning framework, as the distributed and heterogeneous human resources available in crowdsourcing can easily be employed on labeling the active examples. In our research work the active learning examples presented to the crowd for further labeling correspond to the most relevant examples in terms of potential for performance increase.

C. Social Networks as Crowd-based data Sourcing

The constant growth of platforms like Facebook, Twitter or Instagram asset the unquestionable importance of social media in nowadays people's lives. As users in social networks are highly compelled to share, they have become a potential source of information and are often used for professional advertisement, promoting services or market sensing. As an example, in the case of Twitter, each user is allowed to post messages and to include hashtags, a single word started with the symbol #, in order to classify the content of that message and improve search capabilities [18].

Three major conditions explain the success of social networks and their reliability as source media: (i) the variety of contexts in which they can arise, (ii) their ability to grow naturally; and (iii) their intrinsic capability for fastspread [19].

In social networks people define a group of connections and tend to aggregate themselves in communities based on personal interests, geographic proximity or family relationships. Each one of these users' social network connections is able to comment or *like* the posts, according to their personal preferences. By giving their opinion about a particular emotion expressed on a post or picture, each user feeds a recommendation system, by promoting a particular post to be seen by other user's connections. Social networks are thus feeded with data originated from a crowd, whose members are each user's social network connections. This means that data processed by crowdbased social networks is a valuable source of information to feed recommendation systems, as the correlation of users' suggestions to posts and pictures allow users to read or follow specific users.

An interesting work is presented in [20] where micro-blog services are used as a crowdsourcing platform in order to aggregate wisdom under a limited budget. The authors study the Jury Selection Problem in micro-blogs on two crowdsourcing models, one for altruistic users and the other for incentive-requiring users.

III. PROPOSED APPROACH

In this section we describe the proposed approach to define the scope of a crowd in classification problems. The rationale in our approach is to use crowdsourcing information to improve classification. As referred in section II, crowdsourcing can be considered as non-expert knowledge that can be used in classification scenarios. When such classification scenarios encompass a recommendation system, the crowdsourcing information can become a significant asset.

In this approach we tackle this problem going one step further. Given that crowdsourcing can be used as source of data in classification settings [12], we now evaluate the appropriateness of the scope of such crowds by defining different levels of customization of a crowd to each user.

The rationale behind this approach is that by tuning the crowdsourcing to the crowdtarget we can obtain performance improvements. This reasoning is specially interesting when recommendation scenarios are set, since they encompass more subjectivity and dynamics than regular classification scenarios. Moreover, the possibility of model customization based on user preferences is a classification problem in dynamic environments, as the classification model must adapt to the preferences of each new user.

We propose a similarity measure to determine the closeness of a specific user to each crowd contributor and hence define the appropriate crowd scope. Considering a generic recommendation system, we usually have a large set of ratings for every item. When comparing two users a straightforward

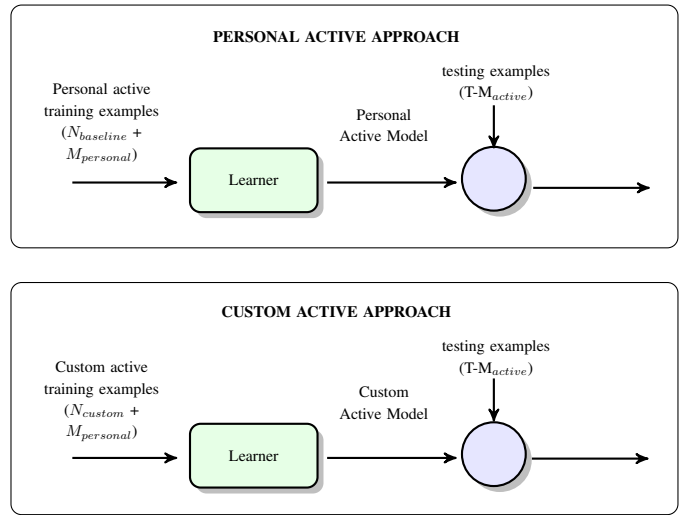


Fig. 1. Personal and Custom Approaches

technique is to use the sum of the absolute differences between items classification.

Taking I as the collection of items, a and b as two different users, the closeness between those users can be estimated by:

$$\sum_{i \in I} |C_i^a - C_i^b|, \quad (1)$$

where C_i^a is the classification of item i given by user a .

A custom crowd suited to provide information for customizing a model to classify documents along with user a preferences, is then a rank of users defined by their closeness to user a provided items classification.

We then use active learning to determine the examples that can provide more information, namely those that are classified with less confidence by the learning machine. The evaluation of this certainty or confidence is dependent on the learning method. An active learning approach is thus based on the idea that the learning algorithm has the ability to choose the learning examples more adequate to the learning process [12], [17], [21], [22].

Considering again a recommendation system, the main idea of using an active subset of examples is to integrate user feedback into the learning process, by either asking each new user or a (customized) crowd to classify those examples to construct the appropriate user profile, just before building the customized model.

Having the active examples selected we propose two setups: A. Personal Active Approach and B. Custom Active Approach that are represented in Fig. 1 and will be detailed in the following sections.

A. Personal active approach

In the personal active approach, the active learning examples are not generically classified, but the user is requested to

classify them, in order to promote the customization. These examples are then named personal active training examples ($M_{personal}$) and removed from the original testing set ($T - M_{active}$). The new user presented to the recommendation system must classify the active learning examples according to his/her preferences, to define his/her user profile. This information is then used along with the remaining training examples to train a personal model.

In this approach the number of personal active training examples has to be necessarily small, since the user is required to provide a classification. Nonetheless, the adequacy of this number is user and task dependent.

Using a recommendation system as example, the active examples are directly classified by the user instead of being classified by the crowd, providing an obvious advantage in terms of representativeness of the training data set.

B. Custom active approach

In the SVM custom active approach we take the strategy one step ahead. On one hand we use the active examples personally classified by the user ($M_{personal}$) in the previous approach, but we also customize the baseline examples that were used so far. To achieve this customization, instead of using the crowd contribution to determine the classification of the baseline examples ($N_{baseline}$), we choose a customized crowd, i.e. a crowd with closer preferences to our target user, resulting in a customized set of examples (N_{custom}).

As already stated, the new user profile is defined by his/her classification in the active learning examples, therefore the closeness between individuals can just take into account the classification of this subset. The baseline training examples classified by this customized group from the crowd is referred as custom examples (N_{custom}).

The main idea behind this approach is to use not only the classification of the user, but tuning the baseline training examples by restricting the contribution of the previously seen individuals to those that are closely related to the new user. The underpinning idea is that the information provided by them can be more valuable, as it avoids the bias provided by using remarkably different users when compared to the one we intent to customize our model for. It is also important to refer that this also avoids asking the new user to manually classify the whole training examples, specially when it is sometimes impossible or unfeasible to ask for such contribution.

IV. EXPERIMENTAL SETUP

In this section we present the experimental setup for both levels of crowd customization described in the previous section. We will describe the recommendation system used as case study and then proceed to introduce the learning mechanism,

detailing the pre-processing steps and the performance metrics used to evaluate the proposed framework.

A. Case study: Humor Recommendation System

Humor research in computer science has two main research areas: humor generation [23], [24] and humor recognition [25], [26], [27]. With respect to the latter, research carried out so far considers mostly humor in short sentences, like *one-liners*, that is jokes with only one line sentence, and the improvement of interaction between applications and users.

This scenario is usually casted as a recommendation system where a user perception of humor has to be grasped/learned so that the system can further recommend jokes that the user finds relevant, fun or interesting.

Humor classification is intrinsically subjective. Each one of us has its own perception of fun, hence automatic humor recognition is a difficult learning task that is gaining interest among the scientific community. Classification methods used thus far are mainly text-based and include diverse classifiers, e.g. SVM classifiers, *naïve Bayes* and decision trees.

In [25] a humor recognition approach based in *one-liners* is presented. A data set was built grabbing *one-liners* from the web using web search engines. This humorous data set was then compared with non-humorous data sets like headlines from news articles published in the Reuters newswire and a collection of proverbs.

In [27] another interesting approach is proposed to distinguish between an implicit funny comment and a not funny one. The authors used a 600,000 web comments data set, retrieved from the Slashdot news Web site. These web comments were tagged by users in four categories: funny, informative, insightful, and negative, which split the data set in humorous and non-humorous comments.

B. Data set

In this work we used the Jester data set as a benchmark. It contains 4.1 million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users and is available at: <http://eigentaste.berkeley.edu>. It was generated from Ken Goldberg's joke recommendation website, where users rate a core set of 10 jokes and receive recommendations from other jokes they could also like. As users can continue reading and rating and most of them end up rating all the 100 jokes, the data set is quite dense.

The data set is provided in three parts: the first one contains data from 24,983 users, the second one from 23,500 users and the third one contains data from 24,938 users. The users from part one and two have rated 36 or more jokes, while the users from the third part have only rated between 15 and 35 jokes. The experiments were carried out using the first two parts as they contain a significant number of users that rate all jokes.

For classification purposes was considered that a joke classified on average above 0.00 is a recommendable joke, and a joke below that value is non recommendable. Jokes were split into two equal and disjoint sets: training and test. The data from the training set is used to select learning models, and the data from the testing set to evaluate performance.

C. Learning model

We will now detail the setup for each approach. Initially, every user, despite his/her profile, contributes equally to the classification of the generically classified examples. Although there are different ways to define the resultant classification based on multiple contributions, like majority voting or weighted voting, we propose a equally weighted voting system based on numeric values. The difference is that a user contributes not only with a binary decision, like liking a book, or not, but also contributes with the corresponding numeric value that scales the likeness ratio.

The learning method used in this experimental setup is the well known and outstanding Support Vector Machine (SVM) [8] with a linear kernel. Given our case study, a joke text classification setup, the SVM is an obvious choice [28]. Nevertheless, the proposed strategies can be applied to any learning algorithm that provides some confidence level in the classification.

New unlabeled examples are classified by the SVM according to which side of the Optimal Separating Hyperplane (OSH) they get into, although not all of them are classified with the same margin to the OSH. Examples close to the margin are those where the SVM puts less confidence, as slight deviations of the OSH would change their given class.

D. Pre-processing

A joke is represented as the most common, simple and successful document representation, which is the vector space model, also known as *Bag of Words*. Each joke is indexed with the *bag* of the terms occurring in it, i.e., a vector with one component for each term occurring in the whole collection, having a value that takes into account the number of times the term occurred in the joke. It was also considered the simplest approach in the definition of term, as it was defined as any space-separated word.

Considering the proposed approach and the use of text-classification methods, pre-processing methods were applied in order to reduce feature space. These techniques, as the name reveals, reduce the size of the joke representation and prevent the mislead classification as some words, such as articles, prepositions and conjunctions, called *stopwords*, are non-informative words, and occur more frequently than informative ones. These words could also mislead correlations

between jokes, so *stopword* removal technique was applied. *Stemming* method was also applied. This method consists in removing case and inflection information of a word, reducing it to the word stem. Stemming does not alter significantly the information included, but it does avoid feature expansion.

E. Performance metrics

In order to evaluate the binary decision task of the proposed models we defined several measures based on the possible outcomes of the classification, such as, error rate ($\frac{FP+FN}{TP+FP+TN+FN}$), recall ($R = \frac{TP}{TP+FN}$), and precision ($P = \frac{TP}{TP+FP}$), as well as combined measures, such as, the van Rijsbergen F_β measure [29], which combines recall and precision in a single score: $F_\beta = \frac{(\beta^2+1)P \times R}{\beta^2P+R}$.

F_β is mostly used in text classification problems with $\beta = 1$, i.e. F_1 , an harmonic average between precision and recall, even though $\beta = 0.5$, i.e. $F_{0.5}$, and $\beta = 2$, i.e. F_2 , can also be used if a specific problem needs to emphasize one metric over the other. While in F_1 precision and recall are evenly weighted, $F_{0.5}$ puts more emphasis on precision than recall, and F_2 weights recall higher than precision. This can easily be seen from the effectiveness measure defined by van Rijsbergen $\epsilon_M = 1 - \frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}}$ with $F_\beta = 1 - \epsilon_M$ where $\alpha = 1/(1 + \beta^2)$ is $0.8(F_{0.5})$, $0.5(F_1)$ and $0.2(F_2)$.

V. EXPERIMENTAL RESULTS AND ANALYSIS

Table I shows the recall and precision results for both levels of crowd customization. We considered that 10 jokes were deemed sufficiently non-intrusive for a user to classify, and the closer crowd in the custom active approach was heuristically defined to have the 1000 users with preferences closer to the user. We can observe that whilst precision values are rather similar, there is a relevant difference of circa 5% in recall values. This difference results in more relevant items being discovered and made available to a user in the recommendation system. Table II shows the F_β comparison between personal

	Precision	Recall
Personal Active Approach	72.78 ± 0.17%	82.69 ± 0.13%
Custom Active Approach	72.35 ± 0.18%	87.32 ± 0.20%

TABLE I
PRECISION AND RECALL PERFORMANCES.

and custom active approaches, with $\beta = 0.5$, $\beta = 1$ and $\beta = 2$. As expected by the difference in recall values, the custom approach presents a better overall performance even when precision gets a higher weight than recall ($\alpha = 0.8$). In spite of the personal active approach only use the 10 active examples it is outperformed by the personal approach that

takes customization one step further by using the similarity measure defined in eq. (1) to choose the crowd that is closer to user preferences.

	$F_{0.5}$	F_1	F_2
Personal Active Approach	74.57	77.42	80.50
Custom Active Approach	74.92	79.13	83.85

TABLE II
 F_β PERFORMANCE RESULTS.

VI. CONCLUSIONS

This paper presented two approaches to make crowds more personal by introducing user profiling information in the definition of classification training data. These approaches were specifically designed to be applied in recommendation systems, where user profiling methods are usually put forward.

The proposed approaches were tested in a humor classification data set and the results obtained showed that making crowds more personal can improve classification results, and ultimately customer satisfaction in recommendation systems.

The reduction of false negatives, revealed by higher values of recall, is directly interpretable as more relevant items being discovered and made available to a user by the recommendation system. One can argue that these relevant items can make the difference in the user's evaluation of the service provided. Still, even when argued that precision is more important when compared with recall in some recommendation system, as users give more relevance to the avoidance of false positives, the customization still provide better results when we combine both metrics under several values of α . These findings are encouraging of research in the topic of crowdtargeting, i.e. suiting crowds to users or tasks with the goal of providing better services. Thus, future work is foreseen in further validating the strategy in different applications of recommendation systems, namely by integrating the approaches in social online scenarios.

REFERENCES

- [1] O. Konstan and J. Riedl, "Deconstructing recommender systems," *IEEE Spectrum*, vol. 10, pp. 1–7, 2012.
- [2] G. Webb, M. Pazzani, and D. Billsus, "Machine learning for user modeling," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 19–29, 2001.
- [3] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.
- [4] J. Surowiecki, *The Wisdom of Crowds*. Doubleday, 2004.
- [5] D. C. Brabham, "Crowdsourcing as a model for problem solving an introduction and cases," *Convergence: the international journal of research into new media technologies*, vol. 14, no. 1, pp. 75–90, 2008.

- [6] C. Silva and B. Ribeiro, "Towards expanding relevance vector machines to large scale datasets," *International journal of neural systems*, vol. 18, no. 01, pp. 45–58, 2008.
- [7] G. Lebrun, C. Charrier, O. Lezoray, and H. Cardot, "Tabu search model selection for svm," *International journal of neural systems*, vol. 18, no. 01, pp. 19–31, 2008.
- [8] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1999.
- [9] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *The Journal of Machine Learning Research*, vol. 5, pp. 255–291, 2004.
- [10] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "The importance of precision in humour classification," in *Proceedings of Intelligent Data Engineering and Automated Learning*, 2011, pp. 271–278.
- [11] —, "Get your jokes right: ask the crowd," in *Proceedings of 1st International Conference on Model and Data Engineering*, 2011, pp. 469–474.
- [12] —, "On using crowdsourcing and active learning to improve classification performance," in *Proceedings of 11th International Conference on Intelligent Systems Design and Applications*, 2011, pp. 469–474.
- [13] S. Greengard, "Following the crowd," *Communications of the ACM*, vol. 54, no. 2, pp. 20–22, 2011.
- [14] J. Barr and L. F. Cabrera, "AI gets a brain," *Queue*, vol. 4, no. 4, pp. 24–29, 2006.
- [15] B. Settles, "Active learning literature survey," CS Technical Report 1648, University of Wisconsin-Madison, 2010.
- [16] C. Silva and B. Ribeiro, "On text-based mining with active learning and background knowledge using svm," *Soft Computing*, vol. 11, no. 6, pp. 519–530, 2007.
- [17] A. K. McCallum and K. Nigam, "Employing EM and pool-based active learning for text classification," in *Proceedings of ICML-98, 15th International Conference on Machine Learning*, 1998, pp. 350–358.
- [18] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "Defining semantic meta-hashtags for twitter classification," in *M. Tomassini et al. (Eds.): ICANNGA 2013, LNCS 7824*. Springer, Heidelberg (2013), 2013, pp. 226–235.
- [19] B. Doerr, M. Fouz, and T. Friedrich, "Why rumors spread so quickly in social networks," *Commun. ACM*, vol. 55, no. 6, pp. 70–75, 2012.
- [20] L. Chen, Y. Tong, C. C. Cao, and J. She, "Whom to ask? jury selection for decision making tasks on micro-blog services," in *Proceedings of the VLDB Endowment*, vol. 5, no. 11, 2012, pp. 1495–1506.
- [21] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.
- [22] S. Dan, "Multi-criteria-based active learning for named entity recognition," Master's thesis, National University of Singapore, 2004.
- [23] O. Stock and C. Strapparava, "Getting serious about the development of computational humor," in *International Joint Conference on Artificial Intelligence*, vol. 18, 2003, pp. 59–64.
- [24] K. Binsted and G. Ritchie, "An implemented model of punning riddles," in *Proceedings of the national Conference on Artificial Intelligence*, 1994, pp. 633–633.
- [25] R. Mihalcea and C. Strapparava, "Technologies that make you smile: Adding humor to text-based applications," *Intelligent Systems*, vol. 21, no. 5, pp. 33–39, 2006.
- [26] —, "Making computers laugh: Investigations in automatic humor recognition," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 531–538.
- [27] A. Reyes, M. Potthast, P. Rosso, and B. Stein, "Evaluating humor features on web comments," in *Proceedings of the International Conference on Language Resources and Evaluation*, 2010, pp. 1138–1141.
- [28] T. Joachims, *Learning Text Classifiers with Support Vector Machines*. Kluwer Academic Publishers, Dordrecht, NL, 2002.
- [29] C. van Rijsbergen, *Information Retrieval*. Butterworths Ed., 1979.