Improved Answer Ranking in Social Question-Answering Portals

Felix Hieber
Department of Computational Linguistics
University of Heidelberg, Germany
hieber@cl.uni-heidelberg.de

Stefan Riezler
Department of Computational Linguistics
University of Heidelberg, Germany
riezler@cl.uni-heidelberg.de

ABSTRACT

Community QA portals provide an important resource for non-factoid question-answering. The inherent noisiness of user-generated data makes the identification of high-quality content challenging but all the more important. We present an approach to answer ranking and show the usefulness of features that explicitly model answer quality. Furthermore, we introduce the idea of leveraging snippets of web search results for query expansion in answer ranking. We present an evaluation setup that avoids spurious results reported in earlier work. Our results show the usefulness of our features and query expansion techniques, and point to the importance of regularization when learning from noisy data.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Question-answering (fact retrieval) systems*

General Terms

Experimentation, Performance

Keywords

Social search and ranking algorithms; Community question answering; Query expansion

1. INTRODUCTION

Community Question-Answering (QA) portals can be characterized as social media that present an alternative to traditional web search. Instead of browsing results of search engines, users present their information needs as detailed questions and get direct responses authored by humans. User-generated language content comes in a high variance in quality: questions and answers range from very high-quality to low quality to irrelevant or even abusive content. This complicates voting for best answers and makes high quality in answer selection all the more important.

The work presented in this paper attempts to construct models and learners that accurately predict answer quality on user-generated data. We base our work on the Yahoo! Answers[1] dataset which has been used in pioneering work by [21] and [22] (henceforth: SCZ) for the task of answer ranking using linguistically motivated features. While more recent work deploys Yahoo! Answers data for complete QA [24, 25], we focus on the aspect of answer ranking, similar to SCZ, leaving aside the important module of question-question matching [12, 11, 17].

The contributions of our work are as follows: Firstly, we present features that explicitly model the quality of answers. Given the information provided in the Yahoo! Answers data released in the Yahoo! Webscope program[2], our quality features implement text-based measures of readability, formality, grammaticality, or entropy. We show in an experimental evaluation that such answer-specific features are a useful complement to question-answer similarity features.

Secondly, our work contributes a new technique for query expansion for answer ranking from noisy data. This is done by applying the idea of piggybacking on web search results to query expansion in answer retrieval. The key idea for our application is to treat each question as a query to a search engine and use the snippets of the top search results as a richer representation of the original. The expansion terms are thus not taken from the set of answers, but outsourced to the richer source of information provided by web search results.

Lastly, we contribute a controlled comparison of ranking models by implementing a perceptron and a ranking SVM in the same stochastic gradient descent (SGD) framework, and by evaluating both models in a clean "answer suggestion" setup that avoids interference from "also good" answers which constitute a problem in the evaluation setup of SCZ. SCZ use in a first step an IR engine to retrieve a set of candidate answers from the full pool of answers for all questions. In a second step the top N answers that were retrieved in the first step are reranked. The ranking problem for a particular question is thus defined as finding the correct answer in the pre-filtered pool of all answers generated for all questions. If a question (or a similar question) has been asked several times by different users in the Yahoo! Answers dataset, different "best answers" are possible, but only one user-selected best answer is considered as correct answer in the test set of SCZ. According to SCZ, selecting spurious best answers caused 18% of the errors in reranking.

[1]`http://answers.yahoo.com`
[2]`http://webscope.sandbox.yahoo.com/`

In our work, we define the ranking problem for a particular question only over the set answers that have been generated by users for this particular question, so that a unique user-voted best answer is guaranteed. We call this evaluation setup "answer suggestion" since it can be thought of as automatic support for the users' voting process for best answers.

2. DENSE MODELS OF STRING SIMILARITY AND QUALITY

The models used in our approach are linear combinations of dense features on string similarity or on string properties instead of on simple word identities. Feature groups 1-3 are reimplementations of features that were shown to be useful in SCZ. We contribute feature group 4 as a feature group that explicitly measures intrinsic textual quality of answers. We disregarded SCZ's "web correlation features" because of lacking access to user click logs and the reported general poor performance of this feature group. Furthermore, we did not compute generalized representations of text (e.g. by n-grams, dependencies, or semantic roles). Instead, all features were computed on full lexical forms.

2.1 Feature group 1: Vector-space similarity

The similarity between a question Q and an answer A is measured by the standard information retrieval metrics of the length-normalized *BM25* formula and the classic *TF-IDF* ranking (see [14]). The actual computation of the features is done with the Terrier[3] platform. Each answer A is considered as a document and added to the Terrier index. The questions constitute the queries. For each question/query Q, Terrier retrieves an ordered list of 3,000 documents/answers for *TF-IDF* and *BM25*, respectively. In our setup, *TF-IDF* and *BM25* scores are extracted only for the n answers that have been posted by users for a question Q. All other metaparameters were set similar to the values reported in SCZ.

2.2 Feature group 2: Word-translation probability

Feature group 2 implements the idea of "bridging the lexical chasm" between questions and answers by using a word-translation model [2, 20]. We used GIZA++[4] to compute word alignments using EM training [4], and smoothed probabilities by a linear interpolation with counts over the whole collection of answers C. The probability $P(Q|A)$ that a question Q is a translation of the answer A is defined in SCZ as follows:

$$P(Q|A) = \prod_{q \in Q} P(q|A) \tag{1}$$

$$P(q|A) = (1 - \lambda)P_{ml}(q|A) + \lambda P_{ml}(q|C) \tag{2}$$

$$P_{ml}(q|A) = \sum_{a \in A} (T(q|a)P_{ml}(a|A)) \tag{3}$$

The probabilities P_{ml} indicate maximum-likelihood estimates, $T(q|a)$ refers to the word translation table computed by GIZA++. The metaparameter λ was set to 0.5 in our experiments.

[3]`http://terrier.org/`
[4]`http://www.fjoch.com/GIZA++.html`

	question	snippet	answer
avg. length	12.8	36.2	59.6

Table 1: Average lengths of questions, snippets, and answers in training set.

2.3 Feature group 3: Textual proximity

Feature group 3 consists of 5 features measuring textual proximity of questions and answers. Each of the 5 features described below contributes 2 values to the feature vector of an answer: First, the raw counts of question terms in the answer, and second the normalized counts for each feature. Normalization is done by dividing the raw count by the question length or by the answer length in the case of *Answer Span.*

Answer span counts the largest distance (in words) between two non-stop question words in the answer.

Informativeness counts the number of non-stop nouns, verbs, and adjectives in the answer that do not appear in the question.

Same word sequence computes the number of non-stop question words that appear in the same order in the answer text.

Overall Match is the number of non-stop question terms found in the answer.

Same sentence match calculates the maximal number of non-stop question terms found in a single sentence in the answer.

2.4 Feature group 4: Textual quality

The fourth feature group implements measures that calculate intrinsic textual quality of answers. These features are inspired by [1]. The question is whether answer-specific features (in contrast to features on pairs of answers and corresponding questions) are informative enough to contribute to answer ranking.

Punctuation measures the number of repeated non-letter characters (including spaces and smilies), the use of capitalization, and the use of html tags.

OOV measures the number of out-of-vocabulary words, i.e., words that do not appear in the top-1000 words in the answer collection.

Readability implements several readability measures calculated from the number of syllables or words in the text and the number of sentences (see [1]).

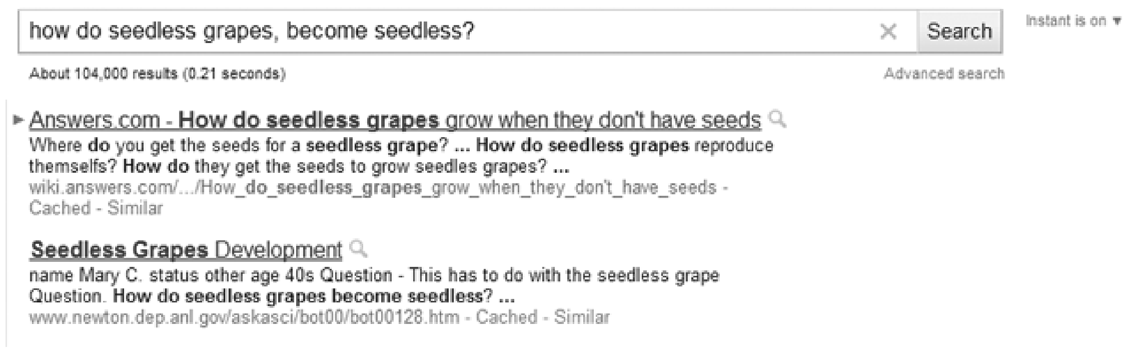
Formality is based on part-of-speech tagged text, and compares the number of "formal" word classes such as nouns, adjectives, prepositions, and articles, against the number of pronouns, verbs, adverbs, and interjections.

Grammaticality counts word n-grams up to length 5 that appear more than 3 times in the collection.

Lastly, *character-level entropy* and *word-level entropy* measure informativeness of the text by calculating the entropy of the character or word distributions.

3. PIGGYBACKING FEATURES ON WEB SNIPPETS

[17] introduced the idea of leveraging web search results to provide greater context for short texts as a way to improve similarity measurements for short search queries. Their idea was evaluated in a setup of query suggestion, however, it can


how do seedless grapes, become seedless?

Search

Instant is on ▾

About 104,000 results (0.21 seconds)

Advanced search

▸ Answers.com - **How do seedless grapes** grow when they don't have seeds

Where **do** you get the seeds for a **seedless grape**? ... **How do seedless grapes** reproduce themselfs? **How do** they get the seeds to grow seedles grapes? ...

wiki.answers.com/.../How_do_seedless_grapes_grow_when_they_don't_have_seeds - Cached - Similar

Seedless Grapes Development

name Mary C. status other age 40s Question - This has to do with the seedless grape Question. **How do seedless grapes become seedless?** ...

www.newton.dep.anl.gov/askasci/bot00/bot00128.htm - Cached - Similar

Figure 1: Google result page for query "how do seedless grapes, become seedless?"

just as well be used for query expansion as in our work. In our work, we will treat a user question as a query to a search engine. We then extend the feature vector of the original question by the same features computed on the snippets of the top search results. This setup resembles the well-known technique of pseudo-relevance feedback [5, 26], however, expansion is not based on the set of top-ranked answers for the original query but instead it is outsourced to the richer set of web snippets for a regular web search using the original query.

The average lengths of questions, snippets, and answers in the training set is shown in Table 1. Query expansion by search result snippets thus clearly is helpful to overcome the length difference between questions and answers. Moreover, our query expansion technique leverages search engine ranking in several ways. Firstly, as shown in [16], parallel data of queries and result snippets provide ideal data to extract synonymous terms for query expansion. Secondly, our approach piggybacks on the information about web link structure that is implicit in the search engine's ranking model.

In our experiments, we piggybacked our query expansion on Google as search engine[5]. Google provides an API to automatically issue queries which returns a maximum of 8 results for each query. We recorded snippets of each result. Figure 1 shows an example snippet on the standard Google search result page for the question *how do seedless grapes, become seedless?*. Interestingly, the first result originates from a Q&A community. Even if the question is not exactly the same, the snippet of the top result introduces terms that are relevant also to the answer of our question, e.g., "seeds", "grape", "grow", "reproduce".

4. LEARNING TO RANK ANSWERS

Since it is rarely possible to classify answers to non-factoid questions as correct or incorrect, the task of learning answer quality is best defined as a ranking task. For a controlled comparison of learners, we do not rely on external software, but present an implementation of a perceptron and a ranking SVM in an SGD framework [3] that is parameterized only in a loss function L. Similar comparisons of perceptron and SVM have been given before[6]. In our case, the relation is made very clear as a difference in regularization, which we evaluate directly in an experimental comparison.

We represent each answer candidate by a feature vector $x_i \in \mathbb{R}^d, i = 1, \ldots, n$, and construct a set of preference pairs P by pairing the best answer with all other answers generated for a question such that $(i, j) \in P$ if x_i is preferred over x_j. The general form of the SGD algorithm for a loss function $L(w) = \sum_{(i,j)\in P} l_{ij}(w)$ defined on pairs is as follows:

```
**for** $t = 1, \ldots, T$ **do**
    Pick at random a preference pair $(i, j) \in P$
    Update $w^{(t+1)} = w^{(t)} - \eta^{(t)} \nabla l_{ij}(w^{(t)})$
**end for**
```

In order to compare the perceptron objective to the SVM objective, it is useful to reformulate the standard loss over misclassified points in the following hinge loss form:

$$l_{ij}(w) = (-\langle w, x_i - x_j \rangle)_+$$

where $(a)_+ = \max(0, a)$. The perceptron update can be understood as a SGD update of this objective, leading to the following update form:

```
**if** $\left\langle w^{(t)}, x_i - x_j \right\rangle < 0$ **then**
    $w^{(t+1)} = w^{(t)} + \eta^{(t)}(x_i - x_j)$
**else**
    $w^{(t+1)} = w^{(t)}$
**end if**
```

The SVM objective can be written in the form of *regularizer + loss* as follows:

$$l_{ij}(w) = \lambda \|w\|^2 + (1 - \langle w, x_i - x_j \rangle)_+$$

Regularization is obtained by minimizing the (squared) ℓ_2-norm of w which is equivalent to finding a hyperplane with large margin. The second term expresses the hinge loss suffered from ranking errors. The regularization metaparameter λ balances the two objectives. Optimizing this objective in a SGD framework yields an update that differs from the perceptron only in the misclassification criterion and the regularization term:

[5]`www.google.com`

[6][8] present a comparison of perceptron, multi-layer perceptron and SVM on the level of objective functions, but not optimization algorithms. [18] present an SGD algorithm for optimizing SVMs in primal form, however, the relation to perceptrons is less clear due to the subgradient projection.

Question/Snippet	Best Answer	Rank
Q: How to remove burnt spot in a stainless pot? S: *but now i have a stainless steel pot that's all black inside.* ***put*** *hot water in your* ***pan*** *and then load the burned spots with* ***baking***	If you **put** a laundry lint sheet in the burnt **pan** with some **water** and boil it, it is suppose the lift the burnt spot off the **pan**. I also use **baking** soda with **water** and boil it, that works too.	3→1 (6)
Q: How do you get your passport renewed? S: *clinton et al. class action lawsuit;* ***check*** *the status of your passport* ***application****. apply for a u.s. passport. apply for a u.s. passport*	You have to send your passport to the passport office along with a renewal **application** and two new photos. [...] **Check** the US passport office site below, you can get the **application** online.	2→1 (6)
Q: How do you say dog in france? S: *for you when you travel to france and other nations where* ***french*** *is spoken. how to say stuff in french "stuff" how to say dog in french* ***"chien"***	Dog, same way you say it in the United States or anywhere else in the world. But in **French** the word for "dog" is **"chien."**	2→1 (8)
Q: how to get dog hair off of **furniture**? S: *remove pet hair from* ***furniture****. how do i get dog hair off* ***furniture****? to remove pet hair from clothing or* ***furniture****, rub article with clean fabric*	There are dense sponges you can buy from **furniture** cleaning companies that can be used to do this. They [...] need to be [...] rubbed over the **furniture**. [...]	3→1 (7)
Q: how to find determent in mXn matrix? S: *transforming a matrix to reduced row echelon form, find the matrix in reduced row echelon form that is row equivalent to the given m x n matrix a. calculate the* ***determinant*** *of the given n x n matrix a. vector spaces*	a **determinant** is not defined for a mxn matrix it's only defined for square matrix	3→1 (4)
Q: How do I fix video games with scrathes? S: *how to fix scratched cds and dis does skipdoctor remove* ***scratches*** *on cds? how to repair a scratched cd, dvd, or video game*	a how bout you put the games back in the case when you are done then they wont have **scratches** on them	5→1 (7)

Table 2: Ranking improvements produced by piggybacking on web snippets.

```
if $\left\langle w^{(t)}, x_i - x_j \right\rangle < 1$ then
    $w^{(t+1)} = w^{(t)} + \eta^{(t)}((x_i - x_j) - 2\lambda w^{(t)})$
else
    $w^{(t+1)} = w^{(t)} + \eta^{(t)}(-2\lambda w^{(t)})$
end if
```

5. EXPERIMENTS

The QA corpus used in our experiments consists of the Yahoo! Answers Manner Questions, version 2.0, as available via Yahoo!'s Webscope program. We split the corpus of 142,627 question-answer pairs into a training set, a development set, and a test set (60%, 20%, 20%) by randomly selecting 85,578 questions for the training set (resulting in 5.76 answers on average), 28,525 questions for development (5.73 answers/question) and 28,524 questions for testing (5.70 answers/question).

For machine learning purposes, each pair of question and correct answer constitutes a positive example, and all other answers for the same question constitute negative examples. This setup was used in training, development, and testing. Note that this setup is different to the evaluation scenario of SCZ where spurious "best answers" coming from related questions can interfere in reranking a pre-selected set of candidate answers chosen from all questions.

As evaluation metrics we used *Average Precision@1* and *Mean Reciprocal Rank*. *Precision@1* is defined as 1 if the correct answer is ranked first, 0 otherwise. *Reciprocal Rank* is defined as the inverse of the rank of the correct answer. The *Average* for both metrics is taken over all questions in the test set. Feature extraction required the implementation of several match-functions for feature groups 3 and 4, and for some features an annotation with part-of-speech tags. We used the Tree Tagger[7] for this purpose. Feature group 1 was outsourced to the Terrier platform. Feature group 2 relies on GIZA++ for calculating word alignments. Web snippets for the construction of larger contexts were based on the Google search engine. We experimented with different numbers of results out of the maximally 8 results returned for each query and found 3 snippets to yield optimal results for the averaged perceptron, while 1 snippet was slightly preferable for the SVM. The learners were implemented as alternative update rules in the same framework, using a decreasing learning rate $\eta^{(t)} = \frac{\eta^{(0)}}{1+t/|P|}$ [23] with $\eta^{(0)} = \frac{2}{|P|}$. The regularization metaparameter λ of the ℓ_2 regularizer of the SVM was adjusted on the heldout data. The optimal value was found to be 10^{-6}. Best results for the perceptron were found by averaging weight vectors over all updates [7]. Parameter averaging did not improve results for the SVM.

[7] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

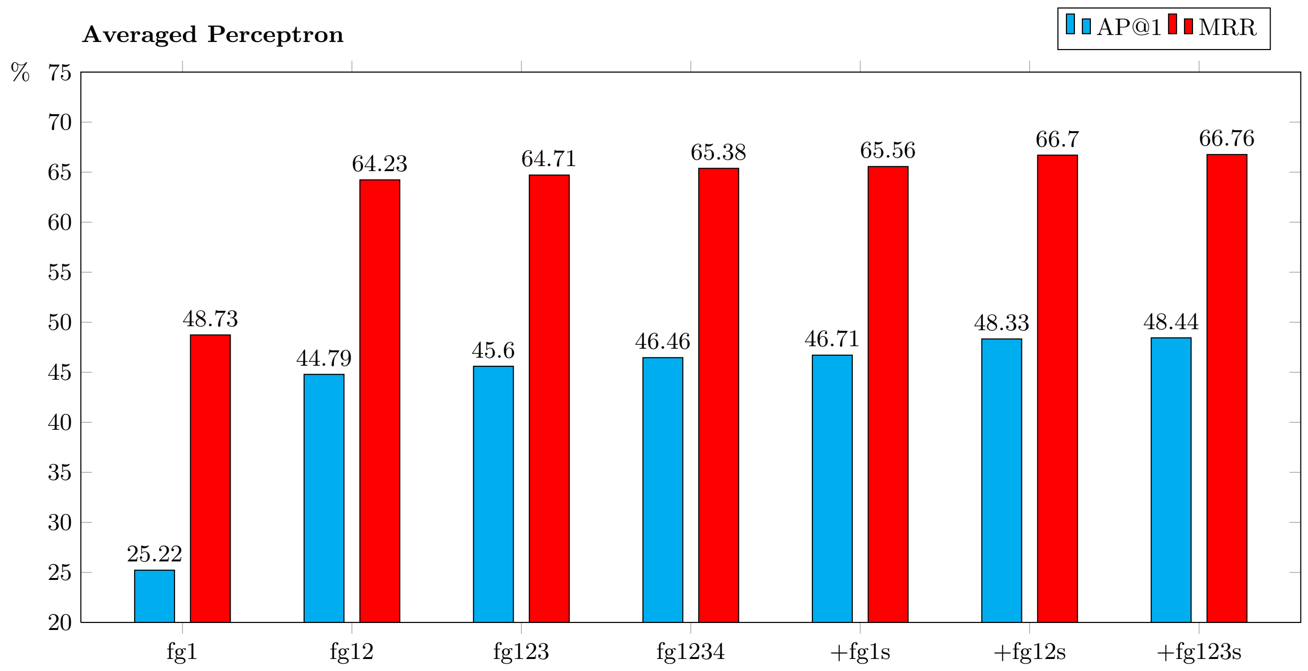


Figure 2: Overall answer ranking results for Averaged Perceptron evaluated with respect to Precision@1 (P@1) and Mean Reciprocal Rank (MRR).

Averaged Perceptron	**fg1**		**fg12**		**fg123**		**fg1234**		**+fg1s**		**+fg12s**	
	AP@1	MRR	AP@1	MRR	AP@1	MRR	AP@1	MRR	AP@1	MRR	AP@1	MRR
fg12	74.57%	31.09%										
fg123	77.76%	32.07%	1.82%	0.75%								
fg1234	81.12%	33.44%	3.74%	1.80%	1.89%	1.04%						
+fg1s	82.08%	33.80%	4.29%	2.07%	2.43%	1.31%	0.53%	0.27%				
+fg12s	88.38%	36.12%	7.90%	3.83%	5.97%	3.07%	4.01%	2.01%	3.46%	1.73%		
+fg123s	88.83%	36.26%	8.16%	3.95%	6.23%	3.17%	4.25%	2.11%	3.71%	1.83%	(0.24%)	0.10%

Table 3: Incremental improvements of feature combinations with fg1 baseline for Averaged Perceptron.

Figures 2 and 3 show the results of the averaged perceptron and the ranking SVM on the test set. As baseline we use a model that includes only feature group 1, thus emulating a standard bag-of-words retrieval model. Feature groups were added incrementally (fg1, fg12, fg123, fg1234). Additionally, feature groups 1-3 are applied incrementally to the snippets, indicated as +fg1s, +fg12s, +fg123s.

Tables 3 and 4 show incremental improvements of feature combinations with fg1 as baseline. Statistical significance was evaluated by an Approximate Randomization test with stratified shuffling at the level of questions [15]. The results for all $\binom{7}{2}$ pairwise comparisons are statistically significant at significance level $p < 0.05$, except for the comparison of +fg123s to +fg12s, and the difference between fg1s to fg1234.

These results can be interpreted as follows: For both learners, we see similar relations with respect to feature combinations. The biggest relative improvement is caused by fg2, the translation feature, confirming SCZ's findings. Smaller improvements can be gained by adding features for textual proximity and quality (fg3 and fg4). Feature computation on web snippets gives another significant gain, again with the biggest boost coming from piggybacking translation features on web snippets.

A comparison of learners shows that the SVM can take better advantage of additional features than the averaged perceptron. Starting from similar baseline results, the SVM yields best results for a combination of all features that are around 2% better than the best results for the averaged perceptron. Given the comparable implementations of learners we can trace this difference back to different regularization mechanisms in the two learners. We found in different experiments that averaging weight vectors improves results by around 2% for the perceptron. However, adjusting λ for optimal performance on the development set improves performance of the SVM by around 2% over the averagedperceptron.

Our results are not directly comparable to SCZ's results, however, it should be noted that our baseline results are lower, and our best results higher than those of SCZ. Furthermore, SCZ could not achieve improvements by using a ranking SVM [10, 13, 6] instead of a simpler perceptron [9, 7, 19]. A possible explanation why similar improvements were not visible in the experiments of SCZ is the use of external software[8] for the SVM computation, which complicates a controlled comparison[9]. Furthermore, our approach of "answer suggestion" provides a clean setup to compare two learners without interference from "also good" answers. Those are unavoidable in SCZ's setup where a pool of all answers generated for all questions is ranked with respect to a particular question.

6. CONCLUSION

Table 2 shows examples for improvements of answer ranking due to piggybacking features onto snippets. The first three examples show how snippets can introduce new terms that are clearly relevant for the best answer. Less exciting but equally effective expansions are shown for the fourth through sixth example. Here either known terms are repeated, and thus boosted, or spelling errors are corrected.

In sum, we can conclude that answer ranking for social QA data is an interesting research field because of the abundance of data and at the same time because of the inherent noisiness of data. In the presented work we have shown that improvements in answer ranking can be achieved from both angles of feature engineering and learning algorithms. In the first case, we have shown that piggybacking feature extraction onto richer resources such as web search results significantly improves matching performance. For the second case, we have shown that appropriate regularization is the crucial ingredient in learning from noisy user-generated data.

The current setup was confined to the task of answer suggestion. This scenario was chosen to provide a clean evaluation setup for a controlled comparison of models and learners. The disadvantage of this scenario is a reduced comparability to the work of SCZ, or standard ranking or retrieval scenarios. For example, a comparison of our web snippet expansion with standard query expansion techniques [26] is not meaningful in an answer suggestion setup where the central problem of low recall is already solved. Future work will address an extension of the described approach to an end-to-end QA system that includes a question-question mapping and a ranking over the full space of answers.

7. REFERENCES

[1] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of WSDM'08*, Palo Alto, CA, 2008.

[2] Adam L. Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*, Athens, Greece, 2000.

[3] Léon Bottou. Stochastic learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, pages 146–168. Springer, Berlin, 2004.

[4] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[5] Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. New retrieval approaches using SMART: TREC 4. In *Proceedings of the Fourth Text REtrieval Conference TREC 4*, Gaithersburg, MD, 1996.

[6] Olivier Chapelle and S. Sathiya Keerthi. Efficient algorithms for ranking with SVMs. *Information Retrieval Journal*, 2010.

[7] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'02)*, Philadelphia, PA, 2002.

[8] Ronan Collobert and Samy Bengio. Links between perceptrons, MLPs, and SVMs. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, Banff, Canada, 2004.

[9] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Journal of Machine Learning Research*, 37:277–296, 1999.

[10] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in large Margin Classifiers*, pages 115–132. MIT Press, Cambridge, MA, 2000.

[11] Jiwoon Jeon, Bruce Croft, and Joon Ho Lee. Finding semantically similar questions based on their answers. In *Proceedings of SIGIR'05*, Salvador, Brazil, 2005.

[12] Jiwoon Jeon, Bruce Croft, and Joon Ho Lee. Finding similar questions in large question and answer archives. In *Proceedings of CIKM'05*, Bremen, Germany, 2005.

[13] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'08)*, New York, NY, 2002.

[14] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] Eric W. Noreen. *Computer Intensive Methods for Testing Hypotheses. An Introduction.* Wiley, New York, 1989.

[16] Stefan Riezler, Yi Liu, and Alexander Vasserman. Translating queries into snippets for improved query expansion. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, Manchester, England, 2008.

[17] Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International World Wide Web conference (WWW'06)*, Edinburgh, Scotland, 2006.

[18] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, Corvallis, OR, 2007.

[19] Libin Shen and Aravind K. Joshi. Ranking and reranking with perceptron. *Journal of Machine Learning Research*, 60(1-3):73–96, 2005.

[8] `http://svmlight.joachims.org/`

[9] [6] also note the possible underperformance of SVMLight due to incomplete training.

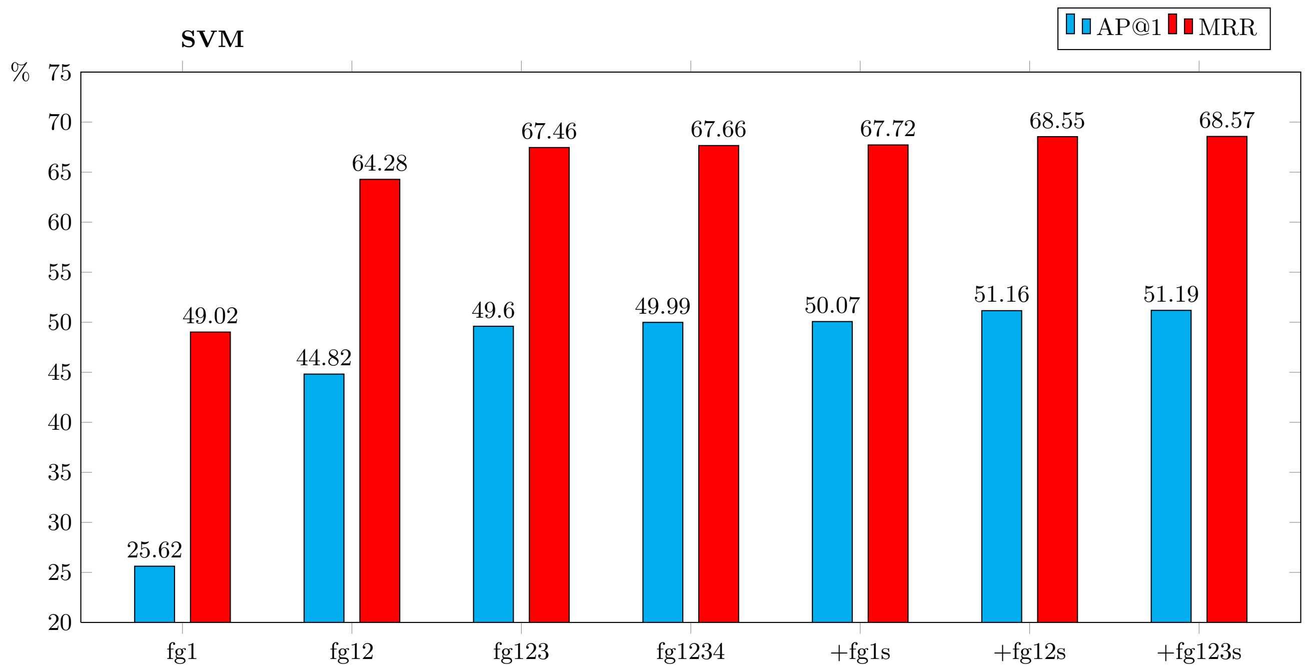


Figure 3: Overall answer ranking results for SVM evaluated with respect to Precision@1 (P@1) and Mean Reciprocal Rank (MRR).

SVM	**fg1**		**fg12**		**fg123**		**fg1234**		**+fg1s**		**+fg12s**	
	AP@1	MRR	AP@1	MRR	AP@1	MRR	AP@1	MRR	AP@1	MRR	AP@1	MRR
fg12	77.33%	31.73%										
fg123	96.24%	38.25%	10.66%	4.94%								
fg1234	97.78%	38.67%	11.53%	5.26%	0.79%	0.30%						
+fg1s	98.12%	38.80%	11.72%	5.36%	0.96%	0.39%	(0.17%)	0.09%				
+fg12s	102.42%	40.49%	14.15%	6.64%	3.15%	1.62%	2.35%	1.31%	2.17%	1.22%		
+fg123s	102.54%	40.54%	14.21%	6.68%	3.21%	1.65%	2.40%	1.34%	2.23%	1.25%	(0.06%)	(0.03%)

Table 4: Incremental improvements of feature combinations with fg1 baseline for SVM.

[20] Radu Soricut and Eric Brill. Automatic question answering using the web: Beyond the factoid. *Journal of Information Retrieval - Special Issue on Web Information Retrieval*, 9:191–206, 2006.

[21] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online QA collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08)*, Columbus, OH, 2008.

[22] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2), 2010.

[23] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP'09)*, Singapore, 2009.

[24] Baoxun Wang, Xiaolong Wang, Chengjie Sun, Bingquan Liu, and Lin Sun. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, Uppsala, Sweden, 2010.

[25] Youzheng Wu and Hisahsi Kawai. Exploiting social q&a collection in answering complex questions. In *Proceedings of the Joint Conference on Chinese Language Processing (CLP2010)*, Beijing, China, 2010.

[26] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, New Orleans, LA, 2001.