RESOLVING DISAGREEMENTS IN MISSPELLING ANNOTATION TASK

Chuchunkov Aleksandr
(alex.chuchunkov@gmail.com)
ITMO University
49 Kronverksky Pr., St. Petersburg, Russia

Shmatova Mariya
(mashashma@yandex-team.ru)
Yandex LLC
16 Lev Tolstoy st., Moscow, Russia

Abstract

For a search engine, it is important to detect and correct spelling errors in queries. Crafting and labeling a golden dataset to train and test a spelling corrector can be a tedious work both in terms of time and human resources. To reduce the labor of an expert (trained linguist), it is possible to delegate the task of labeling spelling errors to less proficient annotators whose labor costs less. In this paper we report the results of experiments on queries to a Russian search engine. We set up the annotation task and evaluated the quality and agreement of annotator submissions. For our experiment, we applied a machine learning classifier to automatically select the best annotation result for each search query in cases of inter-annotator disagreement. The results of these experiments prove that post-annotation stage is required in cases of disagreement among annotators, and that machine learning classifier which resolves the disagreements can perform as a good alternative to a human expert.

Keywords: Human Factors, Measurement, Performance, Machine Learning, Classification

1 Introduction

Spelling correction is a well-established feature of modern search engines. To train and evaluate the performance of such corrector, a golden (labeled) dataset of misspelled queries with respective corrections is required. Due to the importance of the annotation task, it is a good idea to delegate this task to experts (Microsoft Speller Challenge [12] as an example). However, this approach imposes the problems of labor and time cost of the expert annotation. Generally, it takes a lot of time for an expert to label a large dataset of search queries in a thorough manner: only 10% [4] of queries contain spelling errors, thus an expert has to look through at least 10 000 queries to discover 1 000 spelling errors. We set a goal to reduce the cost of labeling task while keeping the high quality of annotation results. First, we decided to utilize workforce of less-proficient annotators (compared to experts mostly having a degree in linguistics). Each query in our dataset was labeled by three annotators independently, and an expert was only presented with queries which raised doubts among the annotators (cases of imperfect agreement). After that, we decided to reduce the labor of an expert even more. With the help of machine learning, we built a classifier which can be used to automatically select the best query correction in cases of disagreement without the help of an expert.

2 Related Work

Annotation of large datasets by means of non-professional annotators has recently become a popular topic for discussion. This approach is time and cost-efficient: the workforce of professional annotators usually costs a

lot. Nowak et al. [8] used workforce of non-professional annotators for picture labeling: the accuracy showed a high agreement of 0.92 which was very close to the agreement among expert annotators. Text annotation by means of multiple non-professionals was described in [5] (Annotating compounds in German), [6] (Annotating named entities in Twitter posts), [13] (Grammatical annotation of Arabic words). Also Gao et al. [7] report about building parallel corpora for machine translation by means of crowdsourcing, i.e. using translations of native speakers instead of those made by professional translators. They also suggest methods of assuring quality of such translations: the work of annotators needs to be evaluated and further tasks should be delegated only to best annotators. To achieve that, the authors used linear classifier to evaluate the translation quality and decide whether it is acceptable (they used both sentence-level and annotator-level features for this classifier). The problem of disagreements in the POS-tagging task is discussed in [10]. During the experiment the annotators were to tag parts of speech for the purpose of POS tagger training. Lower inter-annotator agreement was spotted mainly in "hard" cases (where different linguistic theories imply different tags). The authors used acquired agreement scores to augment the loss function in the learner. Unfortunately, by the time of our experiments there were no mentions of using the workforce of non-professional annotators to correct a set of search queries. Thus we devoted our experiments to applying this approach to the misspelling annotation task.

3 Annotation Task

For the annotation task we used the dataset of 28 631 queries to a Russian search engine. We asked 9 non-professional annotators to look through all the queries and suggest their corrections, each query was given to 3 annotators. After that, the queries with disagreement were annotated by an expert.

3.1 Guidelines

For this task annotators were given guidelines addressing different types of errors in search queries and the ways they should be corrected. The instruction was written by moderators, who analyzed lots of search data before; each correction rule was provided with examples. The guidelines addressed the following types of errors:

- Simple spelling errors ("*elektrolux*" $\rightarrow$ "*electrolux*");
- Word breaking errors ("*der standard.at*" $\rightarrow$ "*derstandard.at*");
- Keyboard layout error (the use of Latin layout for Cyrillic letters or vice versa; "$ищгпре$" $\rightarrow$ "*bought*", "*crblrb*" $\rightarrow$ "$скидки$" ("*discount*"));
- Transliteration errors (the use of Latin alphabet for typing Cyrillic words or vice versa; "*Menya zovut Kxan*" $\rightarrow$ "$Меня\ зовут\ Хан$" ("*My name is Khan*"), "$констракшен$" $\rightarrow$ "*construction*").

Beside easy cases like spelling errors in the common words (such as "*mrket*" instead of "*market*") annotators were to pay attention to more complex cases. For example, they were told to always mind the proper names of companies/products which can be spelled differently: "*Driver Pack*" $\rightarrow$ "*Driverpack*" (without any context both of these queries are correct, but due to the fact that DriverPack is a piece of software, the query needs to be corrected). Such ambiguous cases usually resulted in annotators' disagreement. Another common reason for disagreement was the human factor: anyone can miss an error because of inattention or tiredness due to the volume of the dataset and low percent of misspelled queries.

Moreover, we explicitly stated types of errors which should not be corrected:

- Wrong word forms (e.g. "*italians films*" $\rightarrow$ "*italian films*");
- Punctuation errors (except ones in URLs) (e.g. "*audio technica at2020 usb studio condenser microphone*" $\rightarrow$ "*audio-technica at2020, usb studio, condenser microphone*"; but "*hh/ru*" $\rightarrow$ "*hh.ru*");

- Duplicated words ("*windo ws is is loading files*" $\rightarrow$ "*windows is loading files*").

We asked annotators to preserve the order of words and not to add new words in queries, even if it looks like an idiom or a part of a poem or a song (to be or not be $\rightarrow$ to be or not to be).

We also provided a small mandatory set of queries, which was already annotated by experts. This set was given to annotators before the actual task so that we were sure that guidelines were understood.

3.2 Evaluation

To evaluate the annotator submissions which we obtained as a result of this task, we applied the following set of methods.

- **Accuracy** - ratio of correctly annotated queries to all queries in the test set;
- **Precision** - ratio of correctly annotated queries with spelling errors to all corrected queries;
- **Recall** - ratio of correctly annotated queries with spelling errors to all misspelled queries in the test set;
- **F-measure** - harmonic mean of **Precision** and **Recall**.
- **Spammer Score** (a metric which measures how randomly does the annotator assign labels in a binary labeling task, described in [11]).

We also measured the inter-annotator agreement (IAA) on the dataset level in the following way. For each pair of annotators, we defined inter-annotator agreement as the F-measure where Precision was calculated using submissions of Annotator 1 as the reference, and Recall using submissions of Annotator 2. IAA on the dataset level was calculated as:

$$IAA(corpus) = \frac{\sum\limits_{pairwise} IAA \times \#shared\ queries}{\sum\limits_{pairwise} \#shared\ queries} \tag{1}$$

where $pairwise$ means that we iterate over each pair of annotators (except duplicates with different order). $\#shared\ queries$ stands for the number of queries which were given to both annotators.

3.3 Results

Overall statistics for each annotator are presented in Table 1. It is worth noting that annotators 4 and 9 (A4 and A9) did less work than others with a high percentage of annotation errors. On the contrary, other annotators reached high accuracy (each of them labeled more than 90% of queries correctly).

Table 1: Overall statistics for annotators

	Queries (%)	Acc	Rec	Prec	F-measure	Spammer Score
A1	22.70%	96.85	83.82	84.70	84.26	0.75
A2	36.91%	96.69	84.32	86.00	85.15	0.75
A3	60.57%	95.89	79.43	84.06	81.68	0.68
A4	1.66%	93.46	85.29	76.99	80.93	0.74
A5	29.23%	95.44	81.00	91.40	85.89	0.68
A6	7.08%	95.96	82.84	86.15	84.46	0.71
A7	91.08%	97.15	87.55	87.14	87.34	0.81
A8	49.04%	97.05	86.97	88.41	87.68	0.80
A9	1.74%	89.94	78.00	63.41	69.95	0.64

However, missed errors (judging by Recall) and wrong corrections (judging by Precision) made us believe that annotating each query three times was a reasonable idea. The inter-annotator agreement, average time required to annotate a query and percent of queries for which all three submissions were the same are presented in Table 2.

This task showed that annotation can be delegated to less-experienced workers, and the guidelines worked in general. However, there were cases of annotator disagreement, which still had to be examined and resolved by an expert.

4 Classification Experiment

The annotation task allowed to reduce the labor of an expert, but we wanted to eliminate that labor completely, because an expert was still required for cases of disagreement among the annotators. We decided to conduct an experiment on automatic verification of annotator' submissions to resolve these disagreements. We built a machine learning binary classifier capable of approving/discarding each annotator correction. The results of classification were used to choose a reference out of three submitted corrections.

Table 2: Overall statistics (IAA, time)

IAA	0.85
Time (range), s	3.5 - 27.1
Time (mean), s	14.6
All 3 agreed, %	93.6
Misspelled in queries with disagreement, %	65.6
Misspelled in queries with agreement, %	12.5

4.1 Experimental Setup

For the supervised classification approach, we took the following steps. First, we split the set in 70/30 ratio: the first part was used to train the classifier, and the second part – to test it. We combined a 3-fold cross-validation on the training set with a grid search to find the best parameters (number of estimators, maximum tree depth, maximum number of features, class weights) for Random Forest classifier. Finally, we evaluated classifier performance on the testing set, only taking into account queries with disagreement (Section 4.4).

4.2 Features

We came up with a list of 23 features to train on: 19 features relied on query or correction data (these features were mostly taken from [1]) and 4 features relied on annotation meta data.

4.2.1 Query/correction features

1. Query length (in symbols);
2. Query length (in tokens);
3. Correction length (in tokens);
4. How many tokens were corrected;
5. Query weight in language model;
6. Corrected query weight in language model;
7. Weight difference in language model;
8. Levenshtein distance between corrected query and original query;
9. Weighted Levenshtein distance;
10. Whether the submission agrees with correction by System A, B or both (*Integer*);
11. Whether the submission agrees with the correction by System A (*Boolean*);
12. Whether the submission agrees with the correction by System B (*Boolean*);
13. Whether the correction is different from the query (*Boolean*);
14. Whether the query contains digits (*Boolean*);

15. Whether the query can be interpreted as an URL (*Boolean*);
16. Whether the submission solves the word breaking error (*Boolean*);
17. Whether the submission solves transliteration error (*Boolean*);
18. Whether the submission solves the keyboard layout error (*Boolean*).
19. Whether the submission does not solve neither of these three types of errors (*Boolean*).

4.2.2 Annotation features

20. Time spent on query correction;
21. Average annotation time;
22. Annotator agreement score (ratio of same annotations for given query);
23. Spammer Score of the annotator.

4.3 Classification Method

To classify submissions as wrong or right in cases of disagreement among the annotators, we picked the Random Forest classification algorithm [2], present in Scikit-learn library [9] for Python. To calculate the feature importances and perform feature selection, we used Gini importance algorithm [3], which is the default algorithm for this task in Scikit-learn.

In cases where classifier decided that all submissions for a query are wrong, we did not make any corrections to the query and left it as is.

For the evaluation part we picked Accuracy (*ratio of queries where the submission is the same as the reference among all queries*), Precision (*ratio of queries where errors were correctly annotated among all corrected queries*) and Recall (*ratio of queries where spelling errors were correctly annotated among all originally misspelled queries*).

As a baseline, we evaluated a simple resolving technique - classifier that picks a submission by the majority of votes (*Agreement Select*).

4.4 Results

The results of classification experiment are presented in Table 3 As the table shows, machine learning classifier yielded good results. However, taking one random submission out results in a slightly worse performance.

Feature importance analysis revealed that Random Forests classifier relied mostly on annotation-based features (spent time, annotator agreement score, Spammer Score) and features commonly used in spellcheckers (query length, number of corrections, Levenshtein distance, language model weights).

Table 3: Classifiers performance

2-3 corrections [1777]	**REC**	**PREC**	**ACC**
Agreement select	0.557	0.777	0.646
Random Forests	0.621	0.790	0.686
All queries [8590]	**REC**	**PREC**	**ACC**
Agreement select	0.870	0.947	0.976
Random Forests	0.889	0.946	0.978

5 Conclusions

Our experiments showed that the misspelling annotation task can be done by non-experts, thus reducing the expenses on expert annotation. To evaluate the reliability of non-expert annotation, we applied a set of both well-known (Recall, Precision, Accuracy) and new (Spammer Score) metrics.

To resolve the disagreements among the annotators without the help of an expert, we applied machine learning classification to annotator submissions based on a mixed set of features (query/correction-based and annotation-based). This classifier yielded satisfactory results and it opens the opportunity of omitting human post-editing in the misspelling annotation task.

References

[1] A. Baytin, I. Galinskaya, M. Panina, and P. Serdyukov. Speller performance prediction for query autocorrection. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1821–1824. ACM, 2013.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[4] S. Cucerzan and E. Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *EMNLP*, volume 4, pages 293–300, 2004.

[5] C. Dima, V. Henrich, E. Hinrichs, and C. Hoppermann. How to tell a schneemann from a milchmann: An annotation scheme for compound-internal relations. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1194–1201, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1291.

[6] H. Fromreide, D. Hovy, and A. Søgaard. Crowdsourcing and annotating ner for twitter# drift. *European language resources distribution agency*, 2014.

[7] M. Gao, W. Xu, and C. Callison-Burch. Cost optimization for crowdsourcing translation.

[8] S. Nowak and S. Rüger. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval*, pages 557–566. ACM, 2010.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] B. Plank, D. Hovy, and A. Søgaard. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of EACL*, 2014.

[11] V. C. Raykar and S. Yu. Ranking annotators for crowdsourced labeling tasks. In *Advances in neural information processing systems*, pages 1809–1817, 2011.

[12] K. Wang and J. Pedersen. Review of msr-bing web scale speller challenge. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1339–1340, New York, NY, USA, 2011. ACM.

[13] W. Zaghouani and K. Dukes. Can crowdsourcing be used for effective annotation of arabic? In N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).