A crowdsourced system for user studies in information extraction

Zohreh Khojasteh-Ghamari

Department of Environmental Sciences, Informatics and Statistics,
Ca' Foscari University of Venice,
Via Torino, 155, 30172 Venezia Mestre, Italy
Email: khojaste.z@gmail.com

Abstract: In this paper, from an entity linking (EL) system, we take a set of tweets, where some subsequence of words is annotated with possible meaning/entities and these entities are linked with several Wikipedia pages. We propose a model using crowdsourcing to disambiguate and decide about the accurate Wikipedia page that must be linked with a definite word/spot. We discuss about importance of crowdsourcing and compare different crowdsourcing systems and at the end, introduce crowdflower. We discuss about the crowdflower features in particular. Finally, we analyse output reports of the crowdflower and present a novel approach to select the reliable results. In summary, our observations show that reliable results have a confidence rate over 0.5.

Keywords: crowdsourcing; information extraction; data mining.

Reference to this paper should be made as follows: Khojasteh-Ghamari, Z. (2017) 'A crowdsourced system for user studies in information extraction', *Int. J. Knowledge Engineering and Soft Data Paradigms*, Vol. 6, No. 1, pp.44–51.

Biographical notes: Zohreh Khojasteh-Ghamari received her BS in Information Technology Engineering from Iran in 2009 and MSc in Computer Science from Ca' Foscari University of Venice in 2014. Currently, he is a PhD candidate in Sophia University, Tokyo, Japan.

1 Introduction

The use of crowdsourcing platforms for evaluating the relevance of search results has become a significant strategy (Le et al., 2010). This strategy presents results so quickly with spending trivial amount of money. Apparently, there are still such kind of jobs that involve some element of interpretation, synthesis, or evaluation on which human performs well in contrary to computers poor performance. There are many crowdsourcing channels. The most known is Amazon Mechanical Turk. However, those channels are using only one labour channel. Instead crowdflower combines using 50 labour channel partners. This is also how crowdflower is called Meta crowdsouring system. When a job is uploaded in crowdflower, it distributes the job among the 50 channels and this lead to finish the job in significantly short time. Hence, using crowdflower is like using 50 crowdsouring channel simultaneously. Moreover, crowdflower has an advanced quality control feature which makes us confident about the results, for example it includes hidden questions called Gold Standard.

In Tolomei et al. (2013) applied an entity linking (EL) technique to extract trending entities from a real-world dataset of public tweets. EL is the task of determining the identity of entities which are mentioned in the text. In the other word, it is the task of detecting, in text documents, relevant mentions to entities of a given knowledge base. To this end, entity-linking algorithms use several signals and features extracted from the input text or from the knowledge base. The most important of such features is entity relatedness. A popular choice for EL on open domain text is Wikipedia, and when that is used, the process may be called Wikification [as in the Wikify! (Mihalcea and Csomai, 2007) program, an early EL system].

In general, a typical EL system performs this task in two steps: spotting and disambiguation. The spotting process identifies a set of candidate spots in the input document, and produces a list of candidate entities for each spot. Then, the disambiguation process selects the most relevant spots and the most likely entities among the candidates. Ceccarelli et al. (2013) introduce the Dexter, an open source framework for EL. Dexter implements some popular algorithms and provides all the tools needed to develop any EL technique. Three main methods in Dexter are TAGME (Paolo and Ugo, 2010), the collective linking approach (Han et al., 2011) and WikiMiner (Milne and Witten, 2008). Ceccarelli et al. (2013) proposed a machine learning based approach aimed at discovering the entity relatedness function that can better support the EL task. Orlando et al. (2013) proposed a solution to discover social events from a collection of unstructured press news. Usually there are ambiguities when the text is unstructured. Finin et al. (2010) talked about their experience using both Amazon Mechanical Turk and crowdflower to collect simple named entity annotations for Twitter status updates. As we know Twitter is an informal and abbreviated form in usage of named entity experiment. Moreover, to rigorously address the Twitter EL problem, Guo et al. (2013) proposed a structural SVM algorithm for EL that jointly optimises mention detection and entity disambiguation as a single end-to-end task. Then, they analysed the time series derived from the hourly trending score of each entity as measured by Twitter and Wikipedia. Eventually they noticed that most of the times a poor correlation happened because the trending mention of an entity on Twitter is difficult to disambiguate. Indeed, the EL step mapped this trending mention to the wrong Wikipedia article/entity.

In this paper, as input data, we take two files from the work of Tolomei, et al (2013) and try to disambiguate it by crowdsourcing. One of the files was containing 1,000 tweets, but only 178 of them have one or more spots; and the other was containing 15,623 spots annotated with possible Wikipedia articles. In order to input crowdflower, we combined the two files into a single .tsv file, containing 208 units of work. A unit corresponds to a single occurrence of a spot in a tweet that has more than one Wikipedia page. Hence, there were 208 spots in 178 tweets to disambiguate. The remaining of the paper is as follows, in section 2 we developed a crowdsourced system in crowdflower in order to study user behaviours. In Section 3 we present an evaluation of the outcome and discuss the results. Finally, Section 4 concludes the paper.

2 Design job in crowdflower

In this section, we describe the way we built our job in crowdflower. In order to create a crowdsourcing system in crowdflower, we need to design job, manage quality and finally get the results. Designing job requires uploading data, building job and preview the work. Crowdflower allows uploading file and adding source data via a spreadsheet or pulling data and adding source data via a data feed.

In our work, the input data was made from combination of two files, one was containing 1,000 tweets, but only 178 of them have one or more spots; and the other was containing 15,623 spots annotated with possible Wikipedia articles. We wrote a code to combine these two files. The combination was a tsv file, containing 208 units of work, ready to upload in crowdflower. Each task is a collection of 3 randomly selected units. So we have 70 tasks. We repeat the judgment three times.

For editing a form we used CML Editor which allows us to use code to implement logic and contingencies, HTML, Javascript, and CSS in our forms. Crowdflower uses CML as well as Liquid which is another markup language developed by Shopify, to generate the form needed for each unit of work. Liquid allows to output values into the CML/HTML code from a unit's row in the dataset, by column name (e.g., {{col_name}}). Thus the same CML code is used for all units in the dataset. Crowdflower allows us to use Java Script (w/jQuery) and CSS to further customise the CML. Java Script code is run once on page load knowing that the CML is converted to HTML server side.

Moreover, array data (and JSON objects) cannot be interpreted by Liquid, hence they must be tokenised to strings and then parsed with a Liquid filter (i.e. split). Alternatively JSON objects can be loaded with JS on page load, if the value is written to a form element (as a string) using Liquid. In the survey, we asked workers to identify and reply two questions for each spot with the following guideline: For each tweet, you need to do two jobs:

1 Select the appropriate option by understanding the tweet meaning and eventually the meaning of the underlined words.

2 Rate how much your choice is relevant.

Additionally, in some cases, users use the Graphical Editor since it is easy to work with, but eventually when they need to add some more features then they must change to CML editor. This action is not supported by crowdflower to keep all the data. Therefore, when a user in the middle of work changes the editor from Graphical Editor to CML editor or vice versa, the data will be lost.

2.1 Manage quality

Workers may cheat either because of earning more money but working less or because of their misinterpretation. For this aim, we need to create Test Questions which are known also as Gold set. Test questions are units with known answers that are regularly inserted in the job. They can train contributors and remove underperformers. When some contributors fail too many test questions, system removes them doing with all of their answers from the job. We can say that test questions are the most important quality control mechanism in the crowdFlower platform. In general, these questions are units that the requester has already known the answers and they are inserted in the job quite randomly; therefore, there is almost no way to cheat by a contributor.

Test questions are used two times, once in Quiz Mode which is before a contributor enters to a job, and the other time is during the job.

In each task, contributors answer two questions from the source units and one from the gold units/test questions. For sure contributors are notified if they miss the Gold unit hidden within the page. Furthermore we ask crowdsourcing to do additional judgments if the confidence on one or more task fields is less than the threshold or in other words, the minimum confidence.

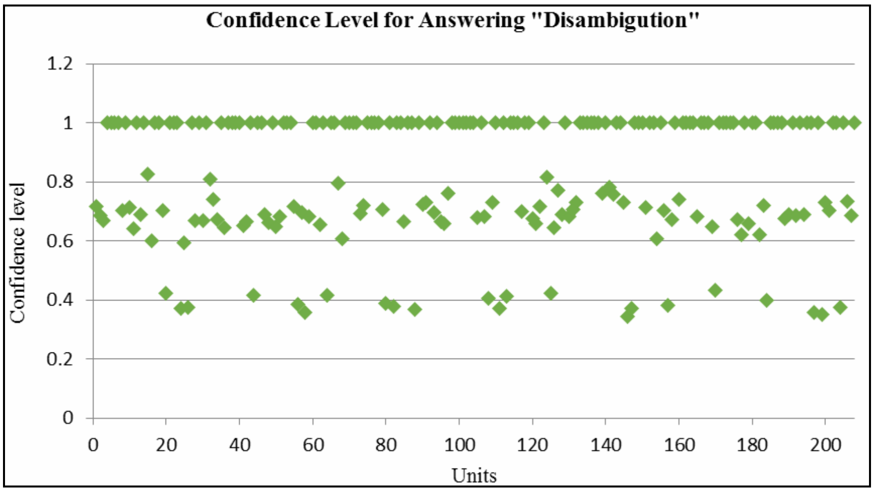
3 Evaluating results

The time required to launch the job to get results (in .CSV format) was about 71 minutes. Completing the job in such short time is the most important advantage of the crowdflower. This comes from the fact that crowdflower distributes the work on 50 labour channels to over five million contributors.

After finishing the task, crowdflower presents us some reports. Two of reports are very important, which are 'all' and 'aggregated' named reports. The former one is named 'full' by crowdflower which returns all unique responses submitted by all trusted contributors for the given field. The result is a newline '\n' delimited list in the aggregated report. 'Full' named report contains the information of all 624 judgments with enough information about each of them. For example, each judgment represented in this file as a row, has different features. For instance, each row has a unique id, the tweet, the spot, possible and suggested options to be voted, and responses of the contributors. In addition, the unit id is the same for a certain unit, so there is one unit id for every three units; because number of judgments are three. Moreover, we can understand the channel in which, a judgment was done. This feature is important for future analysis, because if we find a poor judgment in certain channel, in the next effort we can ban using that channel. Also we can detect country, region, city and even IP of any judgment/contributor.

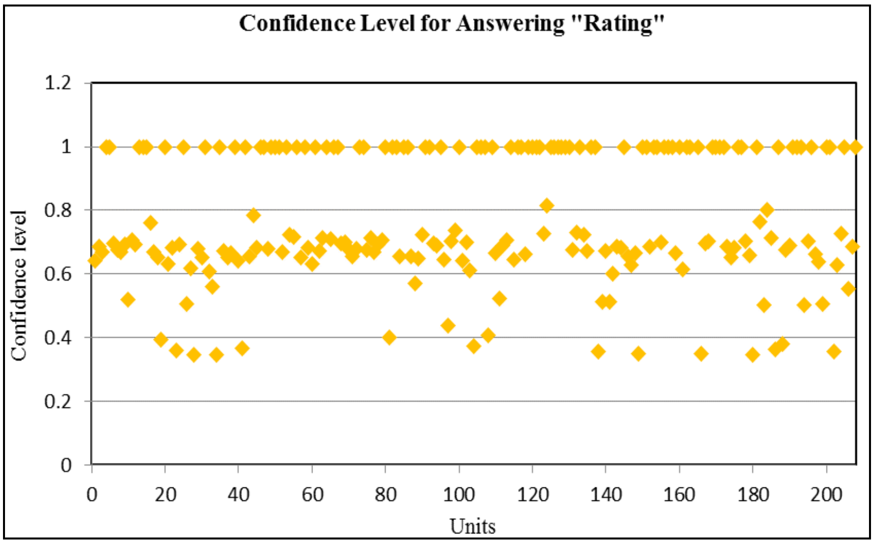
The later file is named 'aggregated'; containing the most useful information, returns a single 'top' result. The 'top' results are also known as the contributor response with the highest confidence (agreement weighted by contributor trust) for the given field. All other responses are ignored. So it works in such a way that in the base of confidence of each judgment, it selects one judgment among three of them, indeed it is selected the one that its contributor has a high confidence from crowdflower. Confidence value is an integer in range of zero to one. This report includes the final vote which is selected from three judgments. In our case, it gives back the Wikipedia page which is selected by crowd along with its relevance. By analysing these two reports, we can reach some important results. The results are depicted in Figures 1 and 2.

Figure 1 Density of disambiguation confidence (see online version for colours)



Note: Average = 0.521.

Figure 2 Density of rating confidence (see online version for colours)

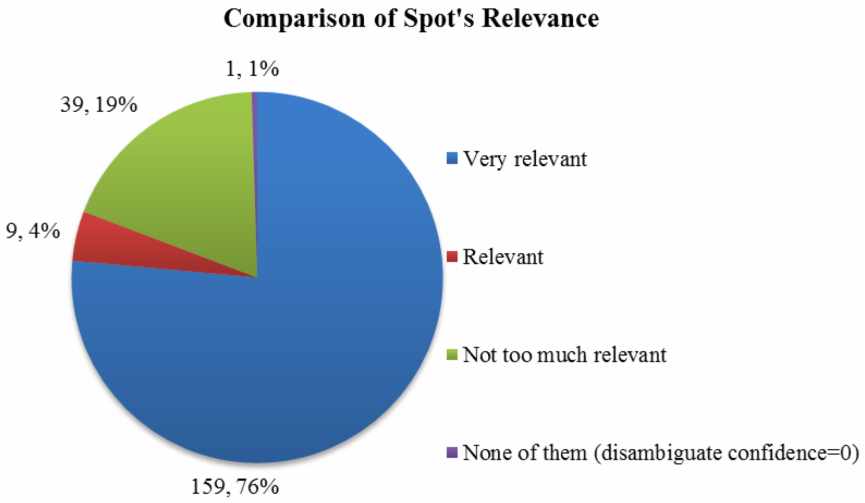


Note: Average = 0.502.

Figure 1 shows the disambiguate results, the average of disambiguation confidence value is 0.521. It is totally depended on the requester how to decide a discarding criteria of the judgments by putting a threshold in confidence. In our case, after analysing the information, we found out that the units with a value of 0.5 or less, represent that all the three judgment's results are different. This situation represents the case that none of the three judges agreed for a unique answer. For example, for a certain question, worker one voted for A, worker two voted for B and worker three voted for C, and at the aggregated report, the one that its worker has high confidence from crowdflower is selected. Since the results from these cases are not reliable, in our work we decided to discard the votes with less than 0.5 of disambiguation confidence. Fortunately, their number is not so high, from 208 units there were only 22 with less than 0.5 disambiguation confidence which represents almost 10% of all the units. Instead there are 109 units with disambiguation confidence of 1, which means that all three judgments for a unit voted for the same answer and it covers 52% of all the units. In fact this kind of report is the most reliable, since all the contributors agreed and compromised in one answer. Number of the rest part which their disambiguation confidence id is between 0.5 and 1, is 77, which is 37% of all and we rely on these judgments too.

Analysing answers of the second part of the question which is "rate" part, we find out that similar to the disambiguation confidence, also here, the number of units with rate confidence less than 0.5 is not so high. Knowing that even if we discard them, it will not be concern. On the contrary of disambiguation confidence, which the number of units with rate confidence 1 was the highest, here the number of units with rate confidence between 0.5 and 1 is the highest.

Figure 3 Comparison of spot's relevance (see online version for colours)



Since there is no grantee about the rules for writing tweets, in the first appearance it seemed that there would be a lot of units with answers of not relevant, but at the end we observed oppositely.

In fact, it is so important that at the beginning how the requester represents the instruction and also the reason of selecting proper answers in Test Questions. If this step goes well, there will be limited number of units which contributors get misunderstanding with them. For instance in our job, according to contributor's idea, our instruction was 78% clear for them. It is undeniable that here are other reasons to lower the confidence too. In our reports the unit with lowest disambiguation confidence is 0.3432. This means that it is completely misunderstood by contributors. Here is the unit (tweet) that is the most misunderstood by contributors:

> Example.1) The worst nicknames in sports – The “Muscle Hamster”, “Molester”, and ”Flying Tomato”? http://t.co/S9R79kL5

For this unit the disambiguate confidence is 0.3432, the rating confidence is 0.6568 and there are five Wikipedia pages to be voted.

On the other hand by analysing the report we find out that the worst (least) rate confidence is 0.3447 with disambiguate confidence of 0.6698 and seven Wikipedia options. The tweet is:

> Example 2: RT @6CancerZone9: No secrets are allowed to be kept from a #Cancer.Thats our job.

There is no limitation about analysing the reports and finding out interesting information, which is absolutely useful for our next works, consequently by changing and modifying the parts that we already got their weakness from previous times, we will reach to more useful achievements.

In our case, it is also important to know how many spots are voted “Not too much 69 relevant”.

The following are tweets with judgment of “Not too much relevant”.

> Example.3) RT @HaydenIsaiah: Don’t act like you like President Obama now since he’s President and you voted Mitt Romney! Mitt Romney was gone have...

For this tweet the disambiguate confidence is 1, rating confidence is 1, and it is with two wikis:{‘prior’: 0.9592592592592593, ‘id’: 426208, ‘title’: ‘Mitt Romney’}

The other tweet is:

> Example.4) Lol got me RT @ theveroniKa: @J Hardaway okay omarion lol

With these information: disambiguate:confidence = 1, rating:confidence = 1,With only one wiki {‘prior’: 1.0, ‘id’: 186260, ‘title’: ‘Omarion’}

One more tweet with this condition is:

> Example.5) RT @JustineLavaworm: For those saying “if Obama wins I’m going to Australia” our PM is a single atheist woman & we have universal he.

With disambiguate confidence = 0.6467, ratingcconfidence = 1, two wikis {‘prior’:0.9652032520325203, ‘id’: 15247542, ‘title’: ‘Atheism’} (two votes) {‘prior’: 0.03219512195121951, ‘id’: 526797, ‘title’: ‘Atheist (band)’} (one vote).

The last point to clarify is that, in our data there were some spots with only one Wikipedia pages. In first appearance, it was so odd to ask crowd to judge a single option question. However, the point is that we have the option of ‘NONE’, which contributors can select it when they could not find the appropriate answer among the suggested options. Knowing this make the judgments for single Wikipedia pages sensible.

4 Conclusions

In this paper we designed a crowdsourced system in crowdflower to solve the ambiguities problem of an EL system. We found out that for our job, the proper crowdsource system is crowdflower which includes 50 labour channel partners. Therefore, the time needed for doing our job is significantly short. Moreover, it contains advanced quality control feature which other crowdsouring systems did not have it. After designing and lunching the job, although we got reports of results, we should decide about selecting the reliable answers. We observed that the output with confidence less than 0.5 should be discarded since these kinds of outputs were the ones with three different answers. Even if in these cases, crowdflower approves one answer whose user has high trust id. In the other hand majority of our outputs, both in disambiguation and rating phase, was with confidence 1 which means all three answers were the same for them. Finally, we succeed to disambiguate more than 90% of the result of an EL system (Tolomei et al., 2013).

Acknowledgements

The author thanks her academic adviser, Professor Salvatore Orlando, for his priceless helps and advices.

References

Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R. and Trani, S. (2013). Dexter: an open source framework for entity linking. In Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval, October, pp.17–20, ACM.

Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R. and Trani, S. (2013) 'Learning relatedness measures for entity linking', In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, October, pp.139–148, ACM.

Crowdflower [online] http://crowdflower.com/ (accessed February 10, 2014).

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J. and Dredze, M. (2010) 'Annotating named entities in Twitter data with crowdsourcing', in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, June, pp.80–88, Association for Computational Linguistics.

Guo, S., Chang, M. W. and Kiciman, E. (2013) 'To link or not to link? A study on end-to-end tweet entity linking', in *HLT-NAACL*, pp.1020–1030.

Han, X., Sun, L. and Zhao, J. (2011) 'Collective entity linking in web text: a graph-based method', in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July, pp.765–774, ACM.

Le, J., Edmonds, A., Hester, V. and Biewald, L. (2010) 'Ensuring quality in crowdsourced search relevance evaluation: the effects of training question distribution', in *Sigir 2010 Workshop on crowdsourcing for Search Evaluation*, July, pp.21–26.

Mihalcea, R. and Csomai, A. (2007) 'Wikify!: linking documents to encyclopedic knowledge', in *Proceedings of the Sixteenth ACM Conference on Conference on information and Knowledge Management*, November, pp.233–242, ACM.

Milne, D. and Witten, I.H. (2008) 'Learning to link with wikipedia', in *Proceedings of the 17th ACM conference on Information and knowledge management*, October, pp.509–518, ACM.

Orlando, S., Pizzolon, F. and Tolomei, G. (2013) 'SEED: A framework for extracting social events from press news', in *Proceedings of the 22nd International Conference on World Wide Web Companion*, May, pp.1285–1294, International World Wide Web Conferences Steering Committee.

Paolo, F. and Ugo, S. (2010) 'TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities)', in *Proceedings of the CIKM.*

Tolomei, G., Orlando, S., Ceccarelli, D. and Lucchese, C. (2013) 'Twitter anticipates bursts of requests for Wikipedia articles', in *Proceedings of the 2013 Workshop on Data-driven User Behavioral Modelling and Mining from Social Media*, pp.5–8, ACM.