Who's who on Gnome mailing lists : identity merging on a large data set

Who's Who on GNOME Mailing Lists: Identity Merging on a Large Data Set

Erik Kouters, Bogdan Vasilescu, Alexander Serebrenik
Technische Universiteit Eindhoven,
Den Dolech 2, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
e.t.m.kouters@student.tue.nl, {b.n.vasilescu, a.serebrenik}@tue.nl

INTRODUCTION

An open-source software project often uses multiple channels of communication (e.g., software repository, mailing list, bug tracker) in each of which an individual has to create a new identity. People might switch between different email addresses (e.g., private and corporate email addresses), causing a mailing list to contain multiple identities for a single individual. Identity merging attempts to solve the challenge of aggregating data by individuals instead of identities (e.g., by merging identities throughout multiple conferences [1]).

Identity merging is the process of identifying which identities belong to the same individual. Aliases are values identifying an individual, commonly found in the form of different $\langle name, emailAddress \rangle$ tuples in mailing lists and source code repositories. By using an identity merging algorithm, two aliases will be matched based on the similarity determined by the algorithm. When two aliases are matched positive, they are considered as belonging to the same individual.

Existing identity merging algorithms are not very robust to noise (e.g., misspelling, diacritics, nicknames, punctuation). We introduce an algorithm which is inspired by information retrieval [2]. We have evaluated this algorithm's performance, and compared it to three existing identity merging algorithms. We present preliminary results of evaluating the performance on a large data set. A priori it is unknown how the algorithms will perform on a data set with this order of magnitude.

The identity merging was performed on GNOME's mailing list archives, which were extracted on April 11, 2012. At that time, the mailing list archives contained $2,202,746$ emails which were sent by $73,920$ distinct email addresses. A previously studied data set, GNOME's software repository logs, was smaller in an order of magnitude, and is expected to contain less noise [2]. The algorithm we introduced is designed to be more robust to the types of noise found in the mailing list archives than the existing identity merging algorithms, and therefore performs better than these existing algorithms. As the data set grows, more people with the same name will occur in the data; existing algorithms do not take this into account (e.g., aliases containing only a common first name will be matched, generating false positives).

EXISTING ALGORITHMS

We consider three existing identity merging algorithms: Simple algorithm by Goeminne and Mens [3], an algorithm by Bird et al. [4], and our interpretation of Bird's algorithm [2]. After implementing and evaluating performance of our interpretation of Bird's algorithm [2], we acquired Bird et al.'s original code. After evaluating this original code, which we will refer to as *Bird Original*, the results compared to our interpretation of Bird's algorithm were very different. For the sake of completeness, we decided to keep both versions for evaluation.

The simple algorithm bases its merging on the *name* and *email address prefix* using simple heuristics. If two aliases share any of the elements, name or prefix, they are considered as a positive match. Additionally, the algorithm has a threshold *minLen* that filters out short words that would easily match everything together.

Bird's algorithms use more complex heuristics such as splitting the *name* into the first and last names, and comparing names by using the first letter of the first name, concatenated with the last name (e.g., Erik Kouters $\Rightarrow$ ekouters). These rules are used on both the name and the email address prefix. Additionally, the algorithms use the Levenshtein distance similarity threshold, *levThr*, to allow for differences in names (e.g., as a result of misspelling).

Algorithm	Precision	Recall	F-measure
Simple Algorithm	0.35	0.90	0.50
Bird's Algorithm	0.18	0.92	0.30
Bird's Original Algorithm	0.41	0.90	0.56
Kouters' Algorithm	0.67	0.98	0.80

TABLE I
THE RESULTS OF EACH ALGORITHM ON GNOME'S MAILING LIST ARCHIVES.

THE ALGORITHM

We introduce an algorithm inspired by information retrieval that is more robust to noise and larger data sets. The bigger a data set becomes, the more likely two people with the same first/last name occurs. Because of the heuristics used by the existing algorithms, it is expected they scale badly when the data set grows by an order of magnitude.

The data set, consisting of a list of aliases, is transformed to Vector Space Model, essentially creating a term-document matrix. This matrix is then augmented by the Levenshtein distance similarity used by Bird et al., due to name differences as a result misspelling. To add robustness with respect to frequent names, we apply the term frequency – inverse document frequency (tf-idf) model which scales to the most frequent name. Finally, the similarity between the aliases are computed using the cosine similarity.

The algorithm accepts three different parameters: *minLen* filters out short words, similar to the simple algorithm; *levThr* adds robustness with respect to misspelling, similar to Bird et al.'s algorithms; *cosThr* defines the threshold of similarity when two aliases are considered as positive or negative matches.

EVALUATION

The algorithms described in this article have been evaluated on GNOME's mailing list archives. The preliminary results are shown in Table I. The parameters used for the algorithms are $minLen = 2$ for the simple algorithm; $levThr = 0.8$ for Bird et al.'s algorithms; $minLen = 2$, $levThr = 0.75$ and $cosThr = 0.75$ for Kouters' algorithm. The choice of these parameters was based on earlier research [2]. Table I refers to Kouters' algorithm which is the algorithm described in the previous section.

All algorithms have a high recall on GNOME's mailing list archives. Even the simple algorithm is able to achieve a high recall with its simple heuristics, showing that the people using the mailing lists are consistently using names when sending emails. Furthermore, Bird et al.'s algorithms do not have a much higher recall than the simple algorithm, despite the more complex heuristics. Kouters' algorithm has the highest recall of all algorithms, even higher than Bird et al.'s algorithms. Bird et al.'s algorithms base their heuristics on the first and last names. However, in some cultures it is common to have a name with more than two words (e.g., a Spanish name typically has a first name and two surnames). Better handling of these type of names might improve recall for Bird et al.'s algorithms.

The precision is what the algorithms differ in the most. Low precision is caused by a high number of *false positive* matches. As a result of complex heuristics, our interpretation of Bird's algorithm is very sensitive to false positives: People with the same last name and first letter of the first name are matched (e.g., Aaron Smith, Alex Smith). Kouters' algorithm prevents matching on common names by applying the tf–idf model; names that occur often are decreased in value. This characteristic scales well with a larger data set.

FUTURE WORK

The parameters that were used for the preliminary results presented in Table I were chosen based on earlier research that included a large-scale experiment that tested all combinations of parameters. Doing a similar large-scale experiment on the large data set is considered future work; we do not know how the parameters will affect the results on a large data set. Ideally, an identity merging algorithm has one optimal parameter combination that performs best on all data sets.

REFERENCES

[1] B. Vasilescu, A. Serebrenik, M. Goeminne, and T. Mens, "On the variation and specialisation of workload – A case study of the Gnome ecosystem community," *Empirical Software Engineering*, pp. 1–54, 2013.

[2] E. Kouters, B. Vasilescu, A. Serebrenik, and M. G. J. van den Brand, "Who's who in GNOME: Using LSA to merge software repository identities," in *Proc. ICSM*, 2012, pp. 592–595.

[3] M. Goeminne and T. Mens, "A comparison of identity merge algorithms for software repositories," *Science of Computer Programming*, 2011.

[4] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in *Proc. MSR*. ACM, 2006, pp. 137–143. [Online]. Available: http://doi.acm.org/10.1145/1137983.1138016