FLINK: An Educator's Tool for Linking Inaccurate Student Records

Frode Eika Sandnes[0000-0001-7781-748X]

Dept. Computer Science, Oslo Metropolitan University, 0130 Oslo, Norway
frodes@oslomet.no

Abstract. Although many areas within the education sector have been subjected to digitalization, including electronic storage and processing of student information, student-administrative tasks are still often handled manually. One such task is the linking of student inaccurate information from different sources such as the task of aligning teachers grade spreadsheets with standardized exam template spreadsheets. Manual linking of records can be tedious, monotonous, and error-prone, especially in large classes with several hundred students. Although automatic robust record linking is common within other areas such as medicine, there are surprisingly few linking tools aimed at educators. The tool FLINK was therefore developed to assist with this task. FLINK was developed over a period of three years through practical experimentation and testing. This paper presents the rationale for the tool, practical use cases, and key design decisions. The tool provides a simple and flexible link between how educators interact with inaccurate student information in practice on one hand, and how they must relate to inflexible administration tools that require exact formal information on the other.

Keywords: Student administration, Robust record linking, Approximate string matching, Dice, Bigram, Learning management systems

1 Introduction

Many education institutions have undergone radical digitalization transformations during the last decades. Teachers typically use learning management systems to facilitate communication between students and teachers including managing students' coursework and the corresponding feedback. In addition, there may be systems for managing reading lists, and systems of managing exam submissions and grades. Occasionally, such systems are not well-integrated with teachers' typical workflow. Although digital education management is a huge improvement over traditional paper-based regimes, there are still manual bottlenecks where the obvious potential of automatic processing has not been fully harvested.

A typical teacher workflow may involve downloading a batch of coursework submitted by a class of students. Each student will have submitted one or more files. Typically, such files are given a mixture of some automatically generated file names based on information about the student such as their name and student ID. Many teachers have their own personal system to read and grade the coursework offline

This is a post-peer-review, pre-copyedit version of the following conference proceeding: Sandnes, F.E. (2023).
FLINK: An Educator's Tool for Linking Inaccurate Student Records. In: Huang, YM., Rocha, T. (eds) Innovative Technologies and Learning. ICITL 2023. Lecture Notes in Computer Science, vol 14099. Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-40113-8_14

such as writing notes in a word-processor file and maintaining grades in a spreadsheet. Teachers then need to invest a significant amount of effort to cross-link and combine results from different assignments.

Moreover, information provided by students is often highly inaccurate. For example, students may use variations or even different names than what is formally recorded in the administrative systems. Also, the student populations in many countries are becoming increasingly diverse with names from all around the world, using different spelling conventions and locale specific characters.

Many state-of-the-art exam management systems allow teachers to upload their results as excel files. Obviously, these will have to be formatted exactly according to students' registered information to prevent registration errors. Sometimes, the lists comprise anonymized exam numbers, other times they may comprise full names and student numbers or a combination of name and student number. This work rests on the observation that many educators manually link data in such situations. Manually linking student records takes valuable time away from other important tasks. More importantly, manual linking is laborious and may lead to fatigue and errors. Clearly, this is especially problematic in classes with several hundred students.

A review revealed that none of the common spreadsheet applications have approximate string matching built in, but some plugins are available such as the commercial flookup plugin for Google Sheets and the fuzzyjoin package for R-project. Microsoft provides a Fuzzy string matcher add-in for Excel. Many education institutions do not allow non-IT staff to install such add-ins due to security policies. Besides these plugins surprisingly few simple tools are available for linking records considering the prominence of and advances in approximate string matching and data linking within the field of computer science. This observation was thus the motivation for this work.

2 Related work

There is a vast literature on approximate string matching going back several decades, and a range of approaches have been proposed [1, 2, 3, 4] as well as string similarity metrics [5, 6].

Gonzalo [2] discusses the general importance of information retrieval systems being tolerant to errors. Chaudhuri et al. [7] addressed inexact database queries while Gravano et al. [8] discussed approximate string joins in web applications. Approximate string matching in the context of record linkage has also received much attention, especially for the integration of large datasets including consensus data [9] and health records [10]. Anonymous approximate record linkage is an active area of research [11, 12, 13, 14, 15] where the goal is to link records without knowing the identity of the individuals described in the records.

Other examples of applications which rely on approximate string-matching techniques include quantitative quality measures for machine translation [16], and text entry acceleration [17].

Traditional approximate string approaches are designed for the Latin alphabets and its various European variations. Such approaches are not effective in other written

languages such as Chinese. One challenge is that the length of a Chinese string is unknown [18]. There are therefore fewer pattern matching algorithms designed for Chinese. One approach is to use traditional pattern matching algorithms with pinyin romanizations [19].

Despite approximate string matching having a long history within computer science and many practical applications in many domains, their practical use seems to be surprisingly uncommon among educators.

3 Student management use cases

Several practical student management use-cases for record linking were identified. These include the following:

1. *Submitting exam results*: Exam systems, such as Inspera and Wiseflow, allow educators to download an exam protocol spreadsheet template to be filled in and uploaded for registration. Teachers do not have to input each individual exam result through the web-interface of the exam systems. Teachers often use a local spreadsheet for determining the final grades in a course based on a portfolio of assignments and/or exam questions. This work often starts before the spreadsheet template becomes available. The template name entries need to be kept exactly as listed. Any discrepancies will result in errors, or the results may not be recorded. The teachers therefore must link their records with the formal identifier used by the system. The current trials were performed using both Inspera and a fagpersonweb (a national system used by public higher education institutions in Norway). The name format for Inspera was “First-names Family names (exam candidate number)” (e.g. “Hank Jones (53)”) and “First-names Family names (student number)” (e.g. “Hank Jones (965472)”). The student candidate number is a running number assigned to each type of exam, while the student number is a unique id to identify a student during the entire studies.
2. *Combining parts*: Usually a course comprises a set of graded assignments. Teachers will typically manage each assignment individually, and later combine the results. This will keep the complexity of each spreadsheet lower while processing each assignment, compared to one large, combined spreadsheet. Moreover, to work with a single spreadsheet for all assignments can be time-consuming as it is necessary to look up the correct row for a given student as it is unlikely the assignments listed in a consistent order. With a single spreadsheet per assignment, the entries can be appended at the end as they are processed. In the current trials, the students had to work on three assignments during the semester, and the results for each assignment were recorded in individual spreadsheets that later had to be combined into one sheet.
3. *Combining records from different systems*: It may be necessary to combine records from several systems that represent student identities differently. With the current trials, the Instructure Canvas learning management system was used during a course for formative assessments, while Inspera was used at the end of a course for summative assessments. Works submitted during a course were labeled using the

following compact form "familynamesfirstnames_*" (e.g. joneshank_79857327_456875) while the "Firstname familyname (number)" format.

4. *Work requirements and pass lists*: Work requirements is a commonly used device whereupon students must satisfy a set of work requirements, such as lab activities, presentations, class attendance, etc., to pass a course. Prior to an exam teachers typically have to submit pass-lists to the exam office that specify which students are allowed to take an exam and which students have not qualified. In the current trials such pass lists, based on obligatory presentations and minimum attendance were submitted using fagpersonweb. The identities of students completing the work requirements (obligatory presentations) were recorded in class. The lists were uploaded electronically using template spreadsheets.
5. *Blacklists*: Pass lists, described above, also need be aligned with the teachers' result lists so that students who have not qualified will not get a grade.
6. *Justification of exam results*: Students sometimes have the right to demand a justification of the exam results. In context of the current trials, students view their exam results in a system known as studentweb. Here, they can also request a justification. The list of justification requests is then electronically forwarded to the teacher. The number of such requests has grown dramatically with these new electronic systems as it simply takes one click to request a justification. Again, the teachers usually will have to consult their spreadsheet records and notes to write the explanations. Clearly, the list of justification requests needs to be linked with the teacher records.

Note that exams are used as a collective term to refer to both traditional exams and portfolio assessment.

4 Name variations

Several classes of variations were observed. Note that these are based on a Norwegian language context. However, the student population is quite diverse with individuals having roots from all over the world. The variations observed will be described to justify the need for a robust linking tool. A summary of common variations is shown in Fig. 1.

Some students may simply submit their coursework labeled with their first name ("Davies") assuming the teacher will know who they are, or just their surname ("Monk"). Some students may also use unofficial nicknames ("Chick") or short forms of their names ("Dick" as in "Richard"). In one case a student used an unofficial first name to match the student's gender identity. Several instances related to problems with surnames were observed, such as using a surname not registered, leaving out part of a surname, and mismatch in the use of hyphenated surname parts. Hyphenated surnames are particularly common in Norway as until recently one was only allowed to be registered with one family name. Individuals with two family names would then have to either treat one of the family names as a first name, or combine the two first names with a hyphen (-).

Name orders can cause confusion, that is, whether the first names come before the surnames or vice versa. In many languages it is possible to differentiate between the first name and surname from the name. This distinction is not straightforward with "Charlie Christian" as an example.

Misspellings can also occur. Although a student is unlikely to misspell their own names, a teacher may introduce spelling mistakes when recording a name from a report into a spreadsheet. Using lowercase characters instead of uppercase in initial caps is not a spelling issue but may cause software mismatches. This can be observed when sorting files in a file explorer. Several such tools treat upper and lower letters as two distinct classes. A similar problem may occur with different character coding such as UNICODE vs ANSI, and the use of ASCII-fied versions of special European characters. Even when the correct character representation is employed some software systems will handle these differently, such as file explorers.

The issues described in the previous sections cause problems when names are matched using exact string-matching algorithms. However, some of these problems are also challenging when linking names manually. This is particularly the case with international names in which a teacher may have limited familiarity, especially languages where certain names are highly frequent.

5 Overview of FLINK

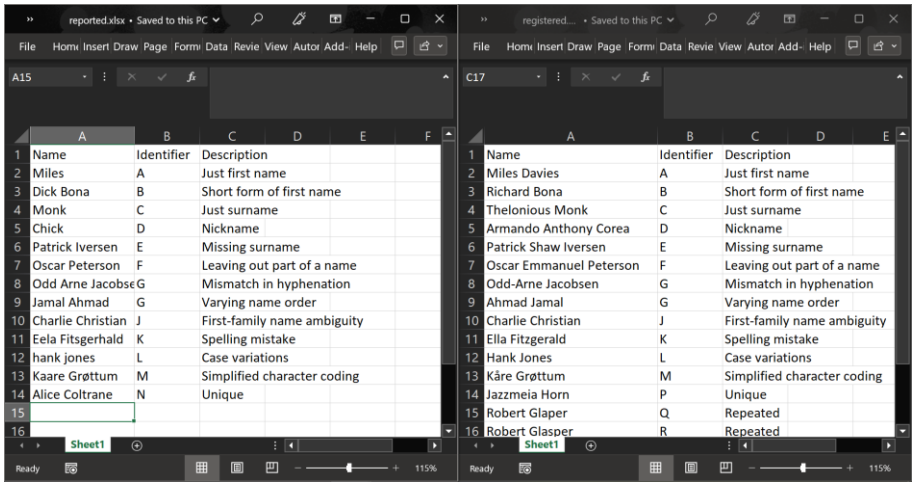
The FLINK tool is designed to be simple and general. FLINK (Fuzzy-LINKer/Frode's LINKer) is an acronym that means "to be good at (studying)" in Norwegian). FLINK takes two spreadsheets as input (see Fig. 1) and provides a linked spreadsheet as output (see Fig. 2). By default, the first columns in the two spreadsheets are used for linking. This is useful if each identifying record for each student is contained within one cell. Alternatively, the user may manually set which columns in each file that should be used for linking the records, for instance if one column contains first name, and another the surname, etc (see Fig. 3).

The output is provided in four sections (see Figs. 4 and 5). The first section contains the successfully linked entries, that is, entries with a successful match from both files. The second section lists entries in the first spreadsheet that were not identified in the second spreadsheet, and the third section lists entries in the second spreadsheet that were not identified in the first spreadsheet. The last section lists potentially duplicate records.

The four sections are intended to help an educator identify errors in the records so that necessary corrective steps can be taken. Often there is a zero tolerance for error as erroneously linked records can have catastrophic consequences for affected students. For instance, a top student could accidentally be recorded as failing a course. The combined results are both displayed and returned as a new spreadsheet.

To further facilitate the prevention of errors each linked record is provided with a color-coded matching score (see Fig. 4), drawing the user's attention towards the records closest to the inclusion threshold. The inclusion threshold can be adjusted.

FLINK is implemented as a JavaScript web application that run locally in the browser without a network connection. It can be used without installation.



	A	B	C
1	Name	Identifier	Description
2	Miles	A	Just first name
3	Dick Bona	B	Short form of first name
4	Monk	C	Just surname
5	Chick	D	Nickname
6	Patrick Iversen	E	Missing surname
7	Oscar Peterson	F	Leaving out part of a name
8	Odd Arne Jacobse	G	Mismatch in hyphenation
9	Jamal Ahmad	G	Varying name order
10	Charlie Christian	J	First-family name ambiguity
11	Eela Fitsgerhald	K	Spelling mistake
12	hank jones	L	Case variations
13	Kaare Grøttum	M	Simplified character coding
14	Alice Coltrane	N	Unique

	A	B	C
1	Name	Identifier	Description
2	Miles Davies	A	Just first name
3	Richard Bona	B	Short form of first name
4	Thelonious Monk	C	Just surname
5	Armando Anthony Corea	D	Nickname
6	Patrick Shaw Iversen	E	Missing surname
7	Oscar Emmanuel Peterson	F	Leaving out part of a name
8	Odd-Arne Jacobsen	G	Mismatch in hyphenation
9	Ahmad Jamal	G	Varying name order
10	Charlie Christian	J	First-family name ambiguity
11	Ella Fitzgerald	K	Spelling mistake
12	Hank Jones	L	Case variations
13	Kåre Grøttum	M	Simplified character coding
14	Jazzmeia Horn	P	Unique
15	Robert Glaper	Q	Repeated
16	Robert Glasper	R	Repeated

Fig. 1. Two spreadsheets with unidentical records

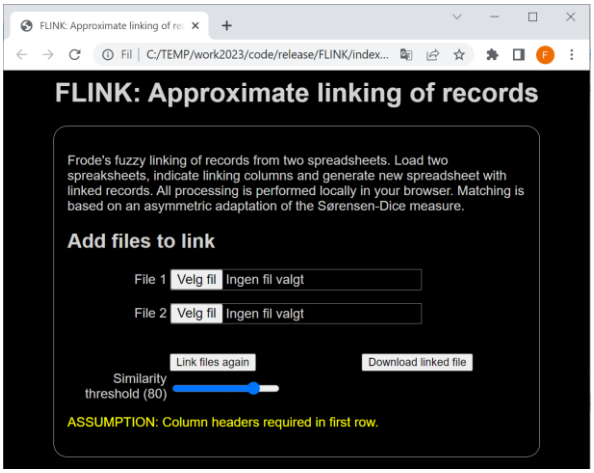


Fig. 2. User interface

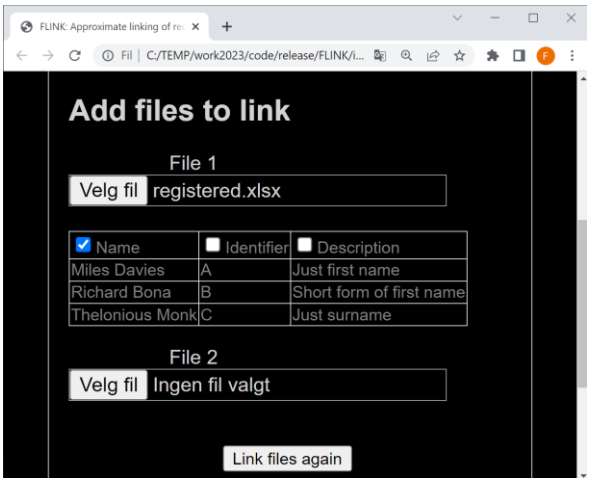


Fig. 3. Specifying linking columns

Matching items

similarity	Name	Name-1	Identifier-1	Description-1	Name-2	Identifier-2	Description-2
1.00	Miles Davies	Miles	A	Just first name	Miles Davies	A	Just first name
1.00	Thelonious Monk	Monk	C	Just surname	Thelonious Monk	C	Just surname
0.93	Patrick Shaw Iversen	Patrick Iversen	E	Missing surname	Patrick Shaw Iversen	E	Missing surname
0.93	Oscar Emmanuel Peterson	Oscar Peterson	F	Leaving out part of a name	Oscar Emmanuel Peterson	F	Leaving out part of a name
0.82	Odd-Arne Jacobsen	Odd Arne Jacobsen	G	Mismatch in hyphenation	Odd-Arne Jacobsen	G	Mismatch in hyphenation
0.73	Ahmad Jamal	Jamal Ahmad	G	Varying name order	Ahmad Jamal	G	Varying name order
0.94	Charlie Christian	Charlie Christian	J	First-family name ambiguity	Charlie Christian	J	First-family name ambiguity
0.67	Ella Fitzgerald	Eela Fitsgerhald	K	Spelling mistake	Ella Fitzgerald	K	Spelling mistake
0.90	Hank Jones	hank jones	L	Case variations	Hank Jones	L	Case variations
0.75	Kåre Grøttum	Kaare Grøttum	M	Simplified character coding	Kåre Grøttum	M	Simplified character coding

Fig. 4. Linking results

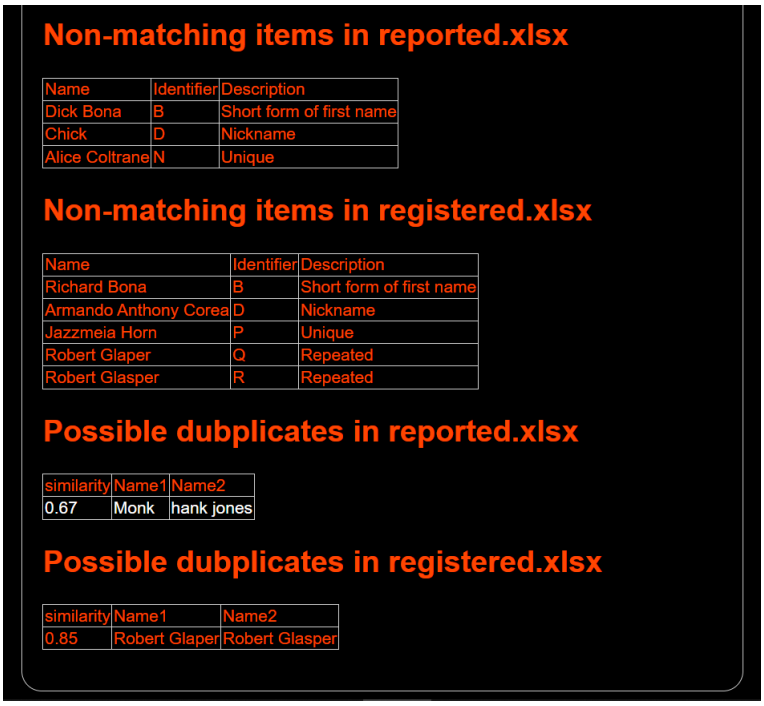


Fig. 5. Problematic record warnings

6 Linking algorithm

The approximate name matching algorithm is based on the well-known Dice-Sørensen distance for text string bigrams, defined as:

$$dice(a,b) = \frac{a \cap b}{2|a||b|} \tag{1}$$

Where a is the set of bigrams for word A and b is the set of bigrams for word B. For example if word A is “hank” with the set of bigrams {“ha”, ”an”, ”nk”}. The score ranges between 1 (complete match) to 0 (completely different). The original Dice-Sørensen measure quantifies the similarity of two texts. However, as information may be missing, an asymmetric modification of the measure was used where the degree with which the sorter string is contained within the longer string, namely:

$$dice_{assymetric}(a,b) = \frac{a \cap b}{\min(|a|,|b|)} \tag{2}$$

Clearly, the strings to be compared are first converted to the same case. Also, for the special cases where the shortest of the two strings does not contain spaces the number of spaces in the longest string are subtracted from the length of the shortest string. This is especially important to ensure matches for short names comprising two

or three characters. For example, if “ho chi min” is matched to “hochimin” we get the two bigram sets {“ho, “o “, “ c”, “ch”, “hi”, “i “, “ m”, “mi”, “in”} (9 bigrams, 2 space characters) and {“ho”, “oc”, “ch”, “hi”, “im”, “mi”, “in”} (7 bigrams, no spaces). Clearly, bigrams with spaces will not match as the corresponding intersection of the two bigram sets is {“ho”, “ch”, “hi”, “mi”, “in”} (5 bigrams). The corresponding similarity is thus $5/(7 - 2) = 1$, that is, full match. Without this adjustment the similarity would just be $5/7 = 0.7$.

Each name is linked with the other name that yields the highest similarity. If the pairing word yields an even higher similarity with another word, the word with the second highest similarity is chosen. This prevents words with high similarities to several other words to be incorrectly matched.

6.1 Experiences

The tool has been used over a period of three years for the use-cases described in Section 3 by the authors. The testing was informal and incremental. FLINK has been adjusted continuously as problems were uncovered. It has reached reasonable stability to be deployed more widely. However, it is the intention to continue to improve the tool as issues are uncovered. There are situations where the tool links incorrectly and it is thus important to manually confirm the results with particular attention towards the matches with the lowest scores. Indeed, there should be a zero-tolerance for errors related to incorrect linking of student records, especially data such as exam results that directly affect the students’ future.

7 Conclusions

Practical record linking needs in context of education and student management was addressed. Several practical use cases were discussed, and problem areas were identified. The web based FLINK linking tool was presented which allows researchers to easily link spreadsheet records for a range of purposes. One implication of this work is that educators spend less time on laborious linking tasks and reduce the risk of errors. The tool and its source can be accessed directly at https://frode-sandnes.github.io/FLINK/.

References

1. Hall, P. A., Dowling, G. R.: Approximate string matching. ACM computing surveys 12(4), 381-402 (1980).
2. Navarro, G.: A guided tour to approximate string matching. ACM computing surveys 33(1), 31-88 (2001).
3. Dorneles, C. F., Gonçalves, R., dos Santos Mello, R.: Approximate data instance matching: a survey. Knowledge and Information Systems 27, 1-21 (2011).
4. Al-Khamaiseh, K., ALShagarin, S.: A survey of string matching algorithms. Int. J. Eng. Res. Appl, 4(7), 144-156 (2014).

5. Winkler, W. E.: String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. Tech report (1990).
6. Gomaa, W. H., & Fahmy, A. A.: A survey of text similarity approaches. international journal of Computer Applications 68(13), 13-18 (2013).
7. Chaudhuri, S., Chen, B. C., Ganti, V., Kaushik, R.: Example-driven design of efficient record matching queries. In: VLDB Vol. 7, pp. 327-338 (2007).
8. Gravano, L., Ipeirotis, P. G., Koudas, N., Srivastava, D.: Text joins in an RDBMS for web data integration. In: Proceedings of the 12th international conference on World Wide Web, pp. 90-101 (2003).
9. Jaro, M. A.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association 84(406), 414-420 (1989).
10. Jaro, M. A.: Probabilistic linkage of large public health data files. Statistics in medicine 14(5-7), 491-498 (1995).
11. Schnell, R., Bachteler, T., Reiher, J.: Privacy-preserving record linkage using Bloom filters. BMC medical informatics and decision making 9(1), 1-11 (2009).
12. Bachteler, T., Schnell, R., Reiher, J.: An empirical comparison of approaches to approximate string matching in private record linkage. In: Proceedings of statistics canada symposium (Vol. 2010). Ottawa, Canada: Statistics Canada (2010).
13. Sandnes, F. E.: HIDE: short IDs for robust and anonymous linking of users across multiple sessions in small HCI experiments. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, (2021).
14. Sandnes, F. E.: CANDIDATE: A tool for generating anonymous participant-linking IDs in multi-session studies. PloS one, 16(12), e0260569 (2021).
15. Sandnes, F. E.: BRIDGE: Administering Small Anonymous Longitudinal HCI Studies with Snowball-Type Sampling. In: INTERACT 2021, LNCS pp. 287-297. Springer International Publishing (2021).
16. Lin, C. Y., Och, F. J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, pp. 605-612 (2004).
17. Sandnes, F. E.: Reflective text entry: a simple low effort predictive input method based on flexible abbreviations. Procedia Computer Science 67, 105-112 (2015).
18. Zhang, L., Zhou, M., Huang, C., Pan, H.: Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 248-254 (2000).
19. Liu, B., Han, D., Zhang, S.: Approximate chinese string matching techniques based on pinyin input method. In: Applied Mechanics and Materials Vol. 513, pp. 1017-1020. Trans Tech Publications Ltd (2014).