Fuzzy and Semantic Search over Encrypted Data in the Cloud

Xiaoyu Zhu[1], Guojun Wang[2](✉), and Dongqing Xie[2]

[1] School of Information Science and Engineering, Central South University,
Changsha 410083, China
zhuxiaoyu@csu.edu.cn

[2] School of Computer Science and Educational Software, Guangzhou University,
Guangzhou 510006, China
csgjwang@gmail.com

Abstract. Cloud computing is becoming more and more popular, many users choose to outsource their data to the cloud. The sensitive data need to be protected before outsourcing, and encryption is usually chosen to protect the data privacy, but it makes the data service such as search a very difficult work. Many searchable encryption schemes are proposed to allow users make effective search over encrypted data. But there is no tolerance of fuzzy and semantic keyword search in one scenario, which greatly affects the search usability. This weakness makes user's searching experiences very low. In this paper, we propose a privacy preserving fuzzy and semantic keyword search scheme over encrypted data in cloud computing. Fuzzy keyword search ability can allow users input search keyword with small typos, and semantic keyword search ability enable returns the matching documents when users' search keyword has the semantic relation with the keywords in the documents. In our scheme, we extract the fuzzy and semantic keyword set using the dictionary and WordNet technique. We also prove that our scheme is privacy preserving through security analysis.

Keywords: Cloud storage · Fuzzy search · Semantic search · Encrypted data · WordNet

1 Introduction

In recent years, cloud storage has becoming more popular than ever as it can allow data owners to store, access and share their data anytime anywhere. Many outsourced data are sensitive and need to be protected from the Cloud Service Provider, which has complete control of the uploaded data. However, for data privacy consideration, data owners usually encrypt the sensitive data before outsourced to the cloud, which makes the deployment of search on encrypted data a difficult task. The simplest way is to download all uploaded data from the cloud and decrypt the encrypted data locally, but the large amount of storage and bandwidth cost makes it an impractical way. Moreover, encryption significantly

G. Wang et al. (Eds.): SpaCCS 2016 Workshops, LNCS 10067, pp. 332–341, 2016.
DOI: 10.1007/978-3-319-49145-5_33

complicates the search operation on the data. Thus, the cloud should explore effective search service on encrypted data while protecting the data privacy.

In order to solve the above problems, many schemes [1–5] have been proposed that provide the search service on encrypted data. Curtmola et al. [2] introduced a SE scheme based on inverted index, and its search process is extremely efficient. In [6–8], the authors proposed ranked keyword search schemes using order-preserving techniques, which can achieve efficient search and meanwhile maintain the accuracy. Boneh et al. [9] proposed the first generalization for symmetric searchable encryption, where data owners can outsource data to the cloud using public key and the data users can search over encrypted data using private key. Goh [10] proposed the first secure index scheme based on bloom filter, which may bring false positive into the result. Scheme [11] proposed a searchable encryption scheme which is not related to the public-key encryption algorithm, and the scheme supports incrementation efficiently. Kamara et al. [12] proposed a dynamic searchable encryption scheme, which supports data insertion and deletion on the encrypted data. Later in scheme [13], they improved their search process by parallelization.

The researchers studied verifiable search functionality extensively in the plaintext database scenario [14,15]. Merkle hash tree is often adopted to verify the search results. But these works only focused on the verification functionality without considering the data privacy. Scheme [6] used hash chain to construct a single keyword search result verification scheme in encrypted data for the first time. Chai et al. [16] proposed a verifiable searchable encryption scheme under a semi-honest-but-curious model. Kurosawa et al. proposed a verifiable searchable encryption scheme against the malicious server in [17], then they extended it to a verifiable dynamic searchable encryption scheme [18], which supports dynamic update operation efficiently.

But almost all these schemes have been designed for exact keyword search. However, in real-life scenarios, the search keywords maybe input with spelling errors or format inconsistencies. The simplest method to implement keyword search over outsourced data is to encrypt all the keywords extracted from the documents. When the data user submits a trapdoor, the cloud server will search the index and return the encrypted document if the trapdoor is in the index list. The main weakness of this scheme is that it only supports exact keyword search.

However, if the users type the wrong keyword, the cloud server will fail to return the search results. Li et al. proposed the first fuzzy searchable encryption scheme using wild-card approach in [19]. This scheme builds the index based on the wildcard technique, which can tolerate the input typos under predefined edit distance. The index is built based on the keywords extracted from the files and the extended keywords generated based on the wildcard technique. But this scheme only supports the search that the input keyword may not exactly match the extended keywords set which includes the extracted keywords and the keywords that are very similar to the extracted keywords. Liu et al. [20] proposed a dictionary based fuzzy searchable encryption scheme with a small index, because the fuzzy keyword set based on wildcard technique contains many meaningless words, this scheme reduces the fuzzy set based on dictionary. In [21], Kuzu et al. proposed a similarity search scheme over encrypted cloud data, the scheme utilized locality sensitive hashing to generate file index. Chuah et al. [22] built the index based on the bedtree technique and implemented the multi-keyword fuzzy search over encrypted cloud data.

However, all the fuzzy searchable encryption schemes mentioned above didn't consider the semantical keywords, for example, "holiday" and "vocation" are semantic related, but they don't have similar structure. Scheme [23] proposed a method which enables semantic keyword search over encrypted documents based on stemming algorithm. In order to support semantic keyword search, the basic way is to generate all the semantically close keywords based on the original keyword.

Scheme [24] proposed a semantic multi-keyword ranked search method, which supports synonym query. Scheme [25] proposed three synonym keyword search schemes and demonstrated the efficiency of their schemes. But these semantic schemes only consider the synonym search and ignore the fuzzy search. Scheme [26] proposed a fuzzy and synonym search scheme, this scheme generates the fuzzy set based on the gram technique and semantic set based on NARCT. However, the gram-based technique still contains many invalid words compared to the dictionary-based technique, and the semantic keyword set only considers the synonyms and ignores the other main semantic relationships. Due to the above drawbacks, the existing schemes signifies that it's important to develop a novel searchable encryption scheme which can support both fuzzy and semantic search, including main semantic relations, not just the synonyms.

We propose a fuzzy and semantic searchable encryption scheme based on dictionary and WordNet, which not only supports privacy preserving fuzzy keyword search, but also provides semantic search over encrypted cloud data, including three main kinds of relations. Fuzzy and semantic keyword search greatly increases the system usability. Our scheme returns all the matching documents when the searching input matches the fuzzy and semantic keyword set. The contributions are summarized as follows:

(1) We utilize the dictionary and WordNet to construct our fuzzy and semantic keyword sets, which reduces the size of the extracted keyword sets. Then we build our secure index using the privacy preserving techniques.
(2) The searching input is expanded based on dictionary and WordNet, the query expansion includes the fuzzy keywords and semantic keywords. The cloud server conducts the search operation and returns all the related files, which greatly improves the system flexibility.
(3) By combining the fuzzy keyword searchable encryption scheme with keyword-based semantic search scheme, we propose a fuzzy and semantic searchable encryption scheme supporting fuzzy and semantic search in one scenario. We prove our scheme is privacy preserving through rigorous analysis.

We organize the rest of this paper as follows. Section 2 introduces the problem formulation. Section 3 describes the details of our proposed fuzzy and semantic search scheme. In Sect. 4, we present the security analysis. The conclusion of our paper is in Sect. 5.

2 Problem Formulation

2.1 System Model

In our fuzzy and semantic keyword searchable encryption scheme, we consider a system comprising a data owner, a data user, and a remote cloud server. The data owner first encrypts a collection of n documents $D = \{d_1, d_2, \cdots, d_n\}$ into a set of ciphertexts $C = \{c_1, c_2, \cdots, c_n\}$, then the data owner outsources the ciphertexts and the index to the cloud. The authorized data user receives a trapdoor from data owner of her search interest via the search control mechanism and then send the trapdoor to the cloud server. Then the cloud server searches over the encrypted index and returns the search result. Figure 1 shows the overall architecture of our scheme.

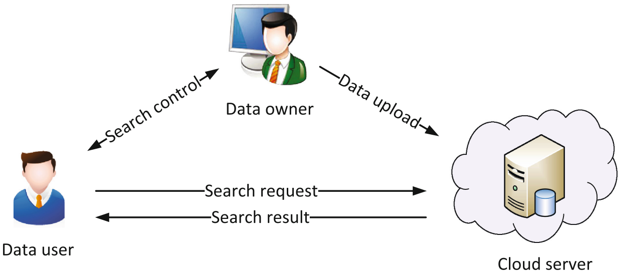

Fig. 1. Architecture of our scheme

2.2 Threat Model

The cloud server is assumed semi-honest-but-curious in this paper, which means that the cloud server may try to derive data privacy from the input trapdoors and search operation. In this work, we aim to propose a privacy-preserving search and protect the sensitive information from the server. While designing our privacy preserving fuzzy and semantic keyword searchable encryption scheme, we adopt the secure definition which are also used in the related work [2]. The cloud server can only access the encrypted files, the secure indexes and the submitted trapdoors. The cloud server can also know and record the search results. We require that the server should not be able to learn any more information.

2.3 Design Goals

In this paper, to enable secure fuzzy and semantic keyword search service over the ciphertexts, our scheme has the following design goals: (1) We propose a method to construct fuzzy and semantic keyword sets based on dictionary and WordNet; (2) We propose a scheme to allow users make fuzzy and semantic search over encrypted data; (3) We prove that our proposed scheme is secure and privacy-preserving.

2.4 Preliminaries

Edit Distance. There are many methods to calculate the string similarity, and we choose edit distance [27] in this paper. $d(w_1, w_2)$ is defined as the edit distance between two strings w_1 and w_2, which is the number of operations required to transform one string into another. There are three primary operations (1) Insertion: insert a character into a string; (2) Substitution: alter a character in a string; (3) Deletion: delete one character from a string.

Dictionary. Dictionary contains a set of legal words. The dictionary is used to reduce the size of the keyword set in this paper. The dictionary excludes all the meaningless English words, and removes all the stop words.

WordNet. WordNet [28] is a large English lexical database. The words are organized into a collection of synonym sets, which represents a lexicalized concept. Synonym sets are linked by different kinds of relations, including synonym, meronym, holonym, hypernym, hyponym and so on. In this paper, we consider three main relations: synonym, meronym/holonym, hypernym/hyponym.

3 Construction of Fuzzy and Semantic Search in Encrypted Cloud Data

3.1 Keyword Set Construction

Fuzzy Keyword Set. The fuzzy keyword search scheme [19] proposed the wildcard technique to generate fuzzy keyword set. The key idea of this scheme is to extract all possible fuzzy keywords under a predefined edit distance, and then build the encrypted index based on fuzzy keyword set. In the wildcard-based fuzzy keyword set construction, $*$ is used as a wildcard, set the edit distance as 1, the keyword set of *use* is $\{use, *use, *se, u*se, u*e, us*e, us*, use*\}$.

The main drawback of scheme [19] is that it pull in all the variations of the keyword, but most of them are invalid. Liu et al. [20] proposed a dictionary based fuzzy searchable encryption scheme with a small index. This scheme uses a dictionary which contains the valid keywords rather than all the variations. For example, considering the wildcard-based fuzzy keyword set of keyword *use*, the fuzzy keywords $*use$ are not valid words, and these meaningless keywords will be removed in this scheme based on dictionary. Thus, the index of [20] is much smaller than the index of [19]. In this paper, in order to reduce the index, we choose the dictionary-based technique to generate our fuzzy keyword set.

Semantic Keyword Set. But all the above fuzzy schemes didn't consider the users' real search intention. Although the two fuzzy schemes extract the keywords and their variations from the documents, but these schemes can only support search with minor typos. If the input keyword doesn't have a similar structure with the exact keyword, the search results will not contain these right documents.

However, there exists one situation, the input keyword may have a semantic relation with the keywords in the documents, so the search scheme should consider the semantic keywords of the original keyword. For example, when the input keyword is "corporation", the fuzzy keyword search scheme will return documents which contain keywords such as "corporation" or "corporations", but it will ignore the semantic keywords such as "firm", "enterprise" and "company". And for some verbs and its variation, such as "bring" and "brought", the fuzzy keyword search scheme didn't consider this scene and will not return the right results. In this paper, we consider three main semantic relations. For example, "firm" is a synonym of "corporation", "wheel" is a meronym of "bicycle" and "flower" is a hypernym of "rose".

In order to cover the ignored semantic keywords of the fuzzy keyword search schemes, the keyword set should consider the fuzzy keywords and the semantic keywords together in one scenario, including three kinds of semantic keyword sets mentioned above.

3.2 Construct Keyword Set

The data owner should first construct the keyword dictionary. The keyword dictionary contains two parts: the fuzzy keyword set and the semantic keyword set. The semantic keyword set contains three parts: synonym set, meronym/holonym set and hypernym/hyponym set.

Firstly, construct the fuzzy keyword set S_w using the dictionary-based method. Then retrieve all the synonyms of the original keyword w from the WordNet and add them into the semantic keyword set set1, then compared the fuzzy keyword set with the synonym set. If the synonym of the keyword is not contained in the fuzzy keyword set, then add it into the keyword set. If the synonym is duplicate, then remove it. The meronym/holonym set and hypernym/hyponym set can be processed in the same way as the synonym set. At Last, our keyword set contains the fuzzy keyword set and semantic keyword set. The keyword set construction is described in Algorithm 1, S_w is denoted as the fuzzy and semantic keyword set of keyword w.

3.3 Encryption

To construct the index for our fuzzy and semantic keyword search scheme, the data owner first generates secret keys (k, sk), where k is a secret key, sk is for algorithm Enc and Dec. Enc is used to encrypt the documents, and Dec is used to decrypt the documents. f is a pseudorandom function. FID_w denotes a set of document IDs whose corresponding documents contain the keyword w.

Our scheme constructs the secure index *Index* by calling the algorithm $\mathsf{SecIndex}$ as follows:

(1) The document set D was encrypted into an encrypted form C by calling the algorithm Enc.

(2) For each keyword $w_i \in W$, construct the fuzzy and semantic keyword set S_{w_i} for the original keyword w_i by calling the algorithm $\mathsf{FuzzySemanticSet}$;
(3) For each keyword $w'_i \in S_{w_i}$, compute the trapdoor set $T_{w'_i} = f(k, w'_i)$;
(4) For each FID_{w_i}, compute the encrypted form $\mathsf{Enc}(k, \text{FID}_{w_i}||w_i)$;
(5) The data owner finally generates the secure index $Index = \{\{T_{w'_i}\}_{w'_i \in S_{w_i}}, \mathsf{Enc}(k, \text{FID}_{w_i}||w_i)\}_{w_i \in W}$;

Finally, the data owner sends the ciphertexts C and the index $Index$ to the server.

Algorithm 1. $\mathsf{FuzzySemanticSet}(w)$

Input: Keyword w.
Output: Fuzzy and semantic keyword set S_w.

```
fuzzy $S_w$= GetFuzzySet();
synonym set1= GetSynonymSet();
for $w'$ in set1 do
    if $w'$ is not in $S_w$ then
        Add $w'$ into $S_w$;
    end if
end for
meronymholonym set2= GetMeronymHolonymSet();
for $w'$ in set2 do
    if $w'$ is not in $S_w$ then
        Add $w'$ into $S_w$;
    end if
end for
hypernymhyponym set3= GetHypernymHyponymSet();
for $w'$ in set3 do
    if $w'$ is not in $S_w$ then
        Add $w'$ into $S_w$;
    end if
end for
return $S_w$
```

3.4 Search Process

When the data user inputs a search keyword w_a, he first generates the fuzzy and semantic keyword set S_{w_a} for the original keyword w_a by calling Algorithm 1, then computes the trapdoors $\{T_{w'_a}\}_{w'_a \in S_{w_a}}$ for $w'_a \in S_{w_a}$. After that, the data user sends the trapdoor set to the remote cloud. Then the server searches $Index$ and returns back all the possible encrypted FIDs as the search result. At last, the data user decrypt the search result using secret key k. If the data user intends to download the documents from the cloud. After receiving the download request from the data user, the server will return the corresponding encrypted documents. At last, the user can decrypt the encrypted documents using the secret key sk.

4 Security Analysis

In this section, we analyze privacy preserving issue for our fuzzy and semantic keyword search scheme.

Theorem 1. The proposed fuzzy and semantic keyword search scheme is privacy preserving.

Proof. In our dictionary based fuzzy and semantic keyword search scheme, we compute the secure index and the trapdoor using the same way. Suppose that there exists an simulator S, a challenger C, it does the followings:

(1) The challenger C sends $|d_1|, \cdots, |d_n|$ and $m = |W|$ to the simulator S.
(2) S keeps (k, sk) secret.
(3) S computes $c_j = \mathsf{Enc}(sk, 0^{|d_j|})$ for $j = 1, \cdots, n$.
(4) S chooses $\{T_{w'_i}\}_{w'_i \in S_{w_i}}$ as the trapdoor set and chooses the secure index $Index' = \{\{T_{w'_i}\}_{w'_i \in S_{w_i}}, \mathsf{Enc}(k, \mathrm{FID}_{w_i}||w_i)\}_{w_i \in W}$ randomly for $i = 1, \cdots, m$.
(5) S sends $C' = (c_1, \cdots, c_n)$ and $Index'$ to C.
(6) S then computes $\{T'_{w'_a}\}_{w'_a \in S_{w_i}}$ as the trapdoor set to C.

In the search phase, C receives(C', $Index'$, $\{T'_{w'_a}\}_{w'_a \in S_{w_i}}$) from S.

Because all the documents are encrypted with CPA-secure algorithms Enc in this scheme, we consider them as confidential to the cloud. Therefore C' and C, $(j, \mathsf{Enc}(sk, 0^{|d_j|}))$ and $(j, \mathsf{Enc}(sk, 0^{|d'_j|}))$ are indistinguishable. $Index'$ and $Index$ are indistinguishable as f is a pseudorandom function. The data privacy is preserved because C cannot learn more information.

5 Conclusion

We propose a fuzzy and semantic keyword searchable encryption scheme while maintaining privacy preserving. We combine the dictionary-based technique and WordNet to build our fuzzy and semantic keyword set. After constructing the keyword set, we further introduce the process of the constructing the secure index, then we propose the search process. Our scheme is an effective solution which enables users make fuzzy and semantic search over encrypted data. We prove that our scheme is privacy preserving through rigorous analysis.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant Numbers 61632009, 61472451 and 61272151, and the High Level Talents Program of Higher Education in Guangdong Province under Funding Support Number 2016ZJ01.

References

1. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceedings of the 2000 IEEE Symposium on Security & Privacy (S&P), pp. 44–55. IEEE (2000)

2. Curtmola, R., Garay, J., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. J. Comput. Secur. **19**(5), 895–934 (2011)
3. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **25**(1), 222–233 (2014)
4. Xia, Z., Wang, X., Sun, X., Wang, Q.: A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **27**(2), 340–352 (2016)
5. Fu, Z., Sun, X., Liu, Q., Zhou, L., Shu, J.: Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans. Commun. **98**(1), 190–200 (2015)
6. Wang, C., Cao, N., Ren, K., Lou, W.: Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Trans. Parallel Distrib. Syst. **23**(8), 1467–1479 (2012)
7. Swaminathan, A., Mao, Y., Su, G.-M., Gou, H., Varna, A.L., He, S., Wu, M., Oard, D.W.: Confidentiality-preserving rank-ordered search. In: Proceedings of the 2007 ACM Workshop on Storage Security and Survivability, pp. 7–12. ACM (2007)
8. Zerr, S., Olmedilla, D., Nejdl, W., Siberski, W.: Zerber+ r: top-k retrieval from a confidential index. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, pp. 439–449. ACM (2009)
9. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24676-3_30
10. Goh, E.-J., et al.: Secure indexes. IACR Cryptology ePrint Archive 2003:216 (2003)
11. Chang, Y.-C., Mitzenmacher, M.: Privacy preserving keyword searches on remote encrypted data. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005). doi:10.1007/11496137_30
12. Kamara, S., Papamanthou, C., Roeder, T.: Dynamic searchable symmetric encryption. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 965–976. ACM (2012)
13. Kamara, S., Papamanthou, C.: Parallel and dynamic searchable symmetric encryption. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 258–274. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39884-1_22
14. Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L.: Dynamic authenticated index structures for outsourced databases. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 121–132. ACM (2006)
15. Pang, H., Tan, K.-L.: Authenticating query results in edge computing. In: Proceedings of the 2004 20th International Conference on Data Engineering, pp. 560–571. IEEE (2004)
16. Chai, Q., Gong, G.: Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In: Proceedings of the 2012 IEEE International Conference on Communications (ICC), pp. 917–922. IEEE (2012)
17. Kurosawa, K., Ohtaki, Y.: UC-secure searchable symmetric encryption. In: Keromytis, A.D. (ed.) FC 2012. LNCS, vol. 7397, pp. 285–298. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32946-3_21

18. Kurosawa, K., Ohtaki, Y.: How to update documents *verifiably* in searchable symmetric encryption. In: Abdalla, M., Nita-Rotaru, C., Dahab, R. (eds.) CANS 2013. LNCS, vol. 8257, pp. 309–328. Springer, Heidelberg (2013). doi:10.1007/978-3-319-02937-5_17
19. Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W.: Fuzzy keyword search over encrypted data in cloud computing. In: Proceedings of the 2010 IEEE International Conference on Computer Communications (INFOCOM), pp. 1–5. IEEE (2010)
20. Liu, C., Zhu, L., Li, L., Tan, Y.: Fuzzy keyword search on encrypted cloud storage data with small index. In: Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 269–273. IEEE (2011)
21. Kuzu, M., Islam, M.S., Kantarcioglu, M.: Efficient similarity search over encrypted data. In: Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, pp. 1156–1167. IEEE (2012)
22. Chuah, M., Hu, W: Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data. In: Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops, pp. 273–281. IEEE (2011)
23. Fu, Z., Shu, J., Sun, X., Zhang, D.: Semantic keyword search based on trie over encrypted cloud data. In: Proceedings of the 2nd International Workshop on Security in Cloud Computing, pp. 59–62 (2014)
24. Metkari, S.S., Sonkamble, S.B.: Multi-keyword ranked search over encrypted cloud data supporting synonym query. Int. J. Sci. Res. **5**(6), 2044–2048 (2016)
25. Moh, T.S., Ho, K.H.: Efficient semantic search over encrypted data in cloud computing. In: International Conference on High PERFORMANCE Computing & Simulation (2014)
26. Jayashri, N., Chakravarthy, T.: Ranked search enabled fuzzy and synonym query over encrypted document in cloud. Int. J. Sci. Eng. Appl. Sci. **1**(8), 215–222 (2015)
27. Levenshtein, V.: Binary codes capable of correcting spurious insertions and deletions of ones. Probl. Inf. Transm. **1**(1), 8–17 (1965)
28. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)