

Stemming

CHRIS D. PAICE
Lancaster University, Lancaster, UK

Definition

Stemming is a process by which word endings or other affixes are removed or modified in order that word forms which differ in non-relevant ways may be merged and treated as equivalent. A computer program which performs such a transformation is referred to as a stemmer or stemming algorithm. The output of a stemming algorithm is known as a stem.

Historical Background

The need for stemming first arose in the field of information retrieval (IR), where queries containing search terms need to be matched against document surrogates containing index terms. With the development of computer-based systems for IR, the problem immediately arose that a small difference in form between a search term and an index term could result in a failure to retrieve some relevant documents. Thus, if a query used the term “explosion” and a document was indexed by the term “explosives,” there would be no match on this term (whether or not the document would actually be retrieved would depend on the logic and remaining terms of the query). The first stemmer for the English language to be fully described in the literature was developed in the late 1960s by Julie Beth Lovins [11]. This has now been largely superseded by the Porter stemmer [14], which is probably the most widely used, and the Paice/Husk stemmer [12]. Stemmers have also been developed for a wide variety of other languages.

Foundations

Definitions

In an IR context, the process of taking two distinct words, phrases or other expressions and treating them as semantically equivalent is referred to as conflation. The two expressions need not be precisely synonymous, but they must refer to the same core concept (compare “computed” and “computable”). In this article, the term “practically equivalent” is used to mean that, for the purposes of a particular application, the words may as well be taken as equivalent. The term conflation is sometimes used as though it is equivalent to stemming, but it is in fact a much broader concept, since it includes (i) cases where the strings concerned are multi-word expressions, as in “access time” and “times for access”, and (ii) cases where the strings are not etymologically related, as in

“index term” and “descriptor”. In case (i) special string matching techniques may be used, whereas in case (ii) reference to a dictionary or thesaurus is necessary. The present account deals exclusively with the conflation of etymologically related single words. There are various possible approaches to word conflation, including the following.

1. Direct matching. In this method, the character sequences of two words are compared directly, and a similarity value is computed. The words are then considered to match if their mutual similarity exceeds a predefined threshold. To give a simple example, the first six letters of the words “exceeds” and “exceeded” are the same, so these words together contain 12 matching letters out of 15. Hence, a similarity of $12/15 = 0.80$ can be computed. Use of a threshold (say, 0.70) allows a decision as to whether the words can be considered equivalent. With such a method, setting the threshold is problematic. Thus, the similarity between “exceeds” and “excess” is 0.62, which is below the stated threshold. However, allowing for this by lowering the threshold to 0.60 would cause “excess” and “except” (similarity 0.67) to be wrongly conflated.

2. Lexical conflation. In this case a thesaurus or dictionary is used to decide whether two words are equivalent. Obviously, this method can be used even for etymologically unrelated words. A problem here is obtaining a suitably comprehensive and up-to-date thesaurus, and one which explicitly lists routine variants such as plurals.

3. Cluster-based conflation. This method, investigated by Xu and Croft [15], involves creating clusters of practically equivalent words by analyzing the word associations in a large representative text corpus. Each query word is then supplemented by adding in the other words in its cluster. In contrast to method (2), the clusters created are specific to the text collection in question. However, the creation of the clusters can be very time-consuming.

4. N-gram conflation. In this method, each word is decomposed into a collection of N-letter fragments (N-grams), and a similarity is computed between the N-gram collections of two words; a threshold is then applied to decide whether the words are equivalent. This approach was pioneered by Adamson and Boreham[1], who used sets of bigrams, where $N = 2$. For example, after eliminating duplicates and sorting into order, “exceeds” can be represented by the bigram set {ce, ds, ed, ee, ex, xc} and “exceeded” by {ce, de, ed, ee, ex, xc}. Out of 7 distinct bigrams here,

5 are shared between the two words; hence a similarity of $5/7 = 0.712$ can be computed.

115 5. Stemming. Stemming refers to the removal of any
suffixes (and sometimes other affixes) from an input
word to produce a stem. Two words are then deemed
to be equivalent if their stems are identical. This
method is much favored because it is fast: all words
120 can be reduced to stems on input to the system, and
simple string matching used thereafter. The remainder
of this article focuses on stemming in this narrow
sense.

125 Prefixes and Infixes

In English, stemmers are usually designed for
removing suffixes from words. The removal of
“intimate” prefixes such as “intro-,” “pro-” and
130 “con-” generally results in words being wrongly
conflated (consider “intro-duction,” “pro-duction”
and “con-duction”).

However, there may be a case for removing looser
prefixes such as “hyper-” or “macro-.” Also, prefix
135 removal may be desirable in certain domains with
highly artificial vocabularies, such as chemistry and
medicine. As explained below, there are some
languages in which removal or replacement of
prefixes, or even infixes, is in fact essential.

140

Performance and Evaluation

Since stemmers were originally developed to aid the
operation of information retrieval systems, it was
145 natural that they were first assessed in terms of their
effect on retrieval performance, as well as on
“dictionary compression” rates. Researchers were
frustrated to find that the effects on retrieval
performance for English language material were small
150 and often negative [10]. Removal of “-s” and other
regular inflectional endings might be modestly helpful,
but use of heavier stemming could easily result in a
loss of performance [7].

Stemming errors are of two kinds: understemming, in
155 which a pair of practically equivalent words are not
conflated, and overstemming, in which two
semantically distinct words are wrongly conflated.

Non-English Stemmers

160

Stemming is appropriate for most (though not all)
natural languages, and appears to be especially
beneficial for highly inflected languages [9]. There is
neither space nor need to describe non-English
165 stemmers here, except to note that some languages
exhibit much greater structural complexity, and this
warrants special approaches. Thus, a typical Arabic

word consists of a root verb of three (or occasionally
four or five) consonants (e.g., “k-t-b” for “to write”),
170 into which various prefixes, infixes and suffixes are
inserted to produce specific variant forms (“katabna”:
“we wrote” and “kitab”:
“book”).

Some researchers have concentrated on extracting the
correct root from a word [3], but Aljlayl and Frieder
175 have demonstrated that better retrieval performance is
obtained by using a simpler “light stemming”
approach, in which only the most frequent suffixes and
prefixes are removed [4]. Their results showed that
extraction of roots causes unacceptable levels of
180 overstemming.

Key Applications

As noted earlier, stemmers are routinely used in
185 information retrieval systems to control vocabulary
variability. They also find use in a variety of other
natural language tasks, especially when it is required to
aggregate mentions of a concept within a document or
set of documents. For example, stemmers may be used
190 in constructing lexical chains within a text. Stemming
can also have a role to play in the standardization of
data for input to a data warehouse.

(Abridged)

195

Recommended Reading

1. Adamson G.W. and Boreham J. The use of an
association measure based on character structure to
200 identify semantically related pairs of words and
document titles. *Inf. Process. Manage.*, 10(7/8):253–
260, 1974.
2. Ahmad F., Yusoff M., and Sembok M.T.
Experiments with a stemming algorithm for Malay
words. *J. Am. Soc. Inf. Sci. Technol.*, 47(12):909–918,
205 1996.
3. Al-Sughaiyer I.A. and Al-Kharashi I.A. Arabic
morphological analysis techniques: a comprehensive
survey. *J. Am. Soc. Inf. Sci. Technol.*, 55(3):189–213,
210 2004.
4. Aljlayl M. and Frieder O. On Arabic search:
Improving the retrieval effectiveness via a light
stemming approach. In , 2002, pp. 340–347.
5. Bacchin M., Ferro N., and Melluci M. A
probabilistic model for stemmer generation. *Inf.*
215 *Process. Manage.*, 41(1):121–137, 2005.
6. Frakes W.B. and Fox C.J. Strength and similarity of
affix removal stemming algorithms. *SIGIR Forum*,
37(1):26–30, 2003 (Spring 2003).
- 220 7. Harman D. How effective is suffixing? *J. Am. Soc.*
Inf. Sci., 42(1):7–15, 1991.

8. Hull D. A Stemming algorithms: a case study for detailed evaluation. *J. Am. Soc. Inf. Sci.*, 47(1):70–84, 1996.
- 225 9. Krovetz R. Viewing morphology as an inference process. *Artificial Intelligence*, 118(1/2):277–294, 2000.
10. Lennon M., Pierce D.S., Tarry B.D., and Willett P. An evaluation of some conflation algorithms for information retrieval. *J. Inf. Sci.*, 3:177–183, 1981.
- 230 11. Lovins J.B. Development of a stemming algorithm. *Mech. Transl. Comput. Linguist.*, 11:22–31, 1968.
12. Paice C.D. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.
- 235 13. Paice C.D. A method for the evaluation of stemming algorithms based on error counting. *J. Am. Soc. Inf. Sci.*, 47(8):632–649, 1996.
14. Porter M.F. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- 240 15. Xu J. and Croft W.B. Corpus-based stemming using co-occurrence of word variants. *ACM Trans. Inf. Syst.*, 16(1):61–81, 1998.

Answer the following questions:

- 1) How would you describe *stemming*? What is its purpose?
- 2) What often resulted in a failure to retrieve relevant documents during searches in the past?
- 3) What is *conflation*?
- 4) Is there any difference between *conflation* and *stemming*?
- 5) What tools have to be used when strings are not etymologically related?
- 6) Describe *direct matching*.
- 7) What does the term of *threshold* refer to in the text?
- 8) What is the disadvantage of *cluster-based conflation*?
- 9) What are *bigrams*?
- 4) *Hyper-* in *hyperactive* is a suffix.
- 5) The term *affix* covers both *prefix* and *suffix*.
- 6) Stemming appears beneficial for highly inflected languages.
- 7) The *light-stemming approach* is based on removing the least frequent affixes.

Match the following terms and their definitions:

- 1) lexical conflation
 - 2) cluster-based conflation
 - 3) N-gram conflation
 - 4) understemming
 - 5) overstemming
-
- a) a method using a corpus of texts
 - b) a method based on bigrams
 - c) a situation where more-or-less equivalent words are not conflated
 - d) a method using a dictionary or thesaurus
 - e) a situation where two semantically distinct words are wrongly conflated

Mark the following statements as *true* or *false*:

- 1) During the conflation, the expressions need to be synonymous.
- 2) The words *mother* and *father* are etymologically related.
- 3) In stemming, two words are considered equivalent provided their stems are identical.

Vocabulary

account [ə'kaʊnt] - výčet; účet

actual ['æktʃu.əl] [-tju-] [-tʃʊl] - vlastní

actually ['æktʃu.ə.li] [-tju-] [-tʃʊ.li] - vlastně

affix [ə'fɪks] - affix (předpona, přípona)

to aggregate st ['æg.gri.gət] - (na)hromadit něco

algorithm ['æl.gə.ri.ðəm] - algoritmus

bigram ['baɪgræm] - bigram (skupina dvou písmen, slabik či slov)

cluster ['kls.tə] - hrozen, skupina, klastr

comprehensive [kəm.pri'hent.sɪv] - komplexní, obsáhlý

conflation [kən'fleɪt] - spojování

compression rate [kəm'preʃ.ən] - kompresivita

consonant ['kɒn.sə.nənt] - souhláska

core [kɔ:r] - jádro; jádřinec

corpus ['kɔ:r.pəs] - korpus, tělo; soubor textů

distinct [dɪ'stɪŋkt] - různý, rozdílný

duplicate ['dju:plɪ.keɪt] - duplikát; duplikovaný (*rovnej výslovnost s „to duplicate“*)

inflectional [ɪn'flekʃənəl] - skloňovací, skloňující, skloňovatelný

equivalent [ɪ'kwɪv.əl.ənt] - ekvivalentní

etymological [et.ɪ'mɒl.ə.dʒi] - etymologický, vztahující se k původu slova

exclusive [ɪk'sklu:zɪv] - výhradní

exhibit st [ɪg'zɪb.ɪt] - vykazovat něco

failure ['feɪ.ljə] - neúspěch

hence [henʃs] - tudíž

however [haʊ'ev.ə] - však, avšak

identical [aɪ'den.tɪ.kəl] - identický, stejný

lexical ['lek.sɪ.kəl] - lexikální

lexical chains ['lek.sɪ.kəl] - lexikální řetězce

loose [lu:s] - volný

mutual ['mju:z.tʃu.əl] - vzájemný

predefined [pri:di'faɪnd] - předem definovaný

prefix ['pri:fxks] - předpona

query ['kwɪə.ri] - dotaz

remainder [ri'meɪn.də] - zbytek

root [ru:t] - kořen

routine [ru:'ti:n] - obvyklý

semantic [si'mæn.tɪk] - sémantický, významový

stem [stem] - kmen; stopka

surrogate ['sʌr.ə.gət] - náhradník; náhradní

suffix ['sʌf.ɪks] - přípona

synonymous [sɪ'nɒn.ɪ.məs] - podobného významu

thereafter [ðeə'ra:f.tə] - poté

thesaurus [θə'so:rəs] - thesaurus

threshold ['θref.həʊld] - práh

thus [ðʌs] - tak, a tak

to aid st [eɪd] - napomáhat něčemu

to arise, arose, arisen [ə'raɪz] - objevit se; vyvstat

to assess st [ə'ses] - hodnotit něco

to conflate [kən'fleɪt] - spojit, spojovat

to decompose st [di:kəm'pəʊz] - rozložit něco

to deem [di:m] - považovat

to duplicate st ['dju:plɪ.keɪt] - duplikovat něco

to eliminate st [ɪ'lɪm.ɪ.neɪt] - eliminovat něco

to exceed st [ɪk'si:d] - překročit něco

to extract st [ɪk'strækt] - extrahovat, vytáhnout něco

to favor st ['feɪ.və] - dávat něčemu přednost

to focus on st ['fəʊ.kəs] - zaměřit se na něco

to investigate st [ɪn'ves.tɪ.geɪt]  - vyšetřovat něco

to merge st [mɜːdʒ]  [mɜːdʒ]  - spojit něco, sloučit

to obtain st [əb'teɪn]  - získat něco

to pioneer [ˌpaɪə'nɪə]  [-'nɪr]  - razit cestu

to retrieve st [rɪ'tri:v]  - vyhledat, vyzvednout něco

to supersede st [ˌsuː.pə'siːd]  [-pə-] 
- nahradit něco

to supplement st ['sʌp.lɪ.mənt]  - doplnit něco

to treat st [tri:t]  - zacházet s něčím

to warrant st ['wɒr.ənt]  ['wɔːr-]  - opravňovat něco

variability ['veə.ri.ə.blɪ]  ['ver.i-]  - variabilita

variant ['veə.ri.ənt]  ['ver.i-]  - varianta

warehouse ['weə.haʊs]  ['wer-]  - skladiště

whereas [weə'ræz]  [wer'æz]  - kdežto

Phrases

In contrast to st - Oproti něčemu

Obviously, ... - Samozřejmě