Methods for Automatic Term Recognition in Domain-Specific Text Collections: A Survey

N. A. Astrakhantsev, D. G. Fedorenko and D. Yu. Turdakov

Haohao Hu, student ID:215448889

Agenda

- Definitions for “term” and “domain”
- Present surveys
- Methods for term recognition
- Efficiency evaluation methods
- Experimental comparisons
- Potential development prospects
- Reference

Definitions for “term” and “domain”

Many definitions of Term from different fields:

- Having analyzed the existing definitions of the term in detail, Pearson concludes that these definitions—particularly, the attempts to separate terms from common words—are based on the assumption that terms can be recognized by intuition.
- To demonstrate the fallacy of this assumption, the so-called “communication attitudes” (in which words can act like terms) are adduced to show that terms are more likely to be used only in some attitudes

Definitions for “term” and “domain” (cont’d)

- Term Features:

  A term can also be defined by its features:

1. Syntactic features: due to the form of the term, e.g. terminological invariance--absence of diversity in writing and pronouncing the term;
2. Semantic features: due to the intention of the term, e.g. intensional exactness--exactness and boundedness of the term meaning;
3. Pragmatic features: due to the specificity of the term behavior, e.g. definiteness—the scientific definition of the term.

Definitions for “term” and “domain” (cont’d)

- Operational definitions of the Term: a word or word combination that denominates a *concept* of a certain field of knowledge or activity.
- How to find out (verify) whether a given concept is specific to a particular domain?

  It is determined by experts in the corresponding domain.

Definitions for “term” and “domain” (cont’d)

- Analyzing only average-specific terms and wide domains:

  1) reducing the requirements for the level of expertise in the domain;

  2) improving the coordination of expert actions;

  3) increasing the effectiveness of applications that use recognized terms.

- The definition of the Term depends on the *application*.

Definitions for “term” and “domain” (cont’d)

- Categories of term recognition scenarios:

  1. According to the interpretation of term frequency:

     (a) considering (classifying) each individual occurrence of the term;

     (b) do not distinguish between occurrences of one term.

  2. According to the number of terms to be recognized:

     (a) recognizing a predetermined number of terms;

     (b) in which the number of terms to be recognized is determined by the algorithm for each input collection.

Definitions for “term” and “domain” (cont’d)

- Categories of term recognition scenarios (cont'd):

3. According to the length of a term candidate:
   - (a) recognizing one-word terms only;
   - (b) recognizing two-word terms only;
   - (c) recognizing multi-word terms only;
   - (d) recognizing terms of any length.

Present surveys

1. One of the first surveys on term recognition [19] analyzes two directions: automatic indexing and term recognition itself.

   a) focused on the TF-IDF methods

   b) introduce the aspects of the term—*unithood* (word relations in multi-word terms) and *termhood* (relatedness of the term to the domain)

   c) analyze term recognition methods according to the aspect which is characteristic of the corresponding method.

   d) separates two classes of methods: linguistic and statistical.

Present surveys (cont'd)

2. M. Pazienza et al. [3], note that the present works regard linguistic methods as sets of filters and do not explicitly distinguish between these classes.

emphasis:

a) word association measures (Dice Factor, *z* test, *t* test, χ^2 test, *MI*, MI^2, MI^3, and likelihood ratio)

b) the simplest methods for determining domain specificity of the term (term frequency, C-value, and co-occurrence).

Methods for term recognition

- General scheme for the scenario that does not distinguish between occurrences of one term:

1. Candidates collection:

   i) linguistic filters: selecting only nouns and nominal groups (word combinations in which the noun is the main word)

   ii) noise filtration: filtering out candidates with the number of occurrences less than 2 or 3, candidates found in a preset stop word list, non-alphabetic symbols and words composed of one letter

2. Computation of features for term candidates

3. Feature-based inference: estimation of the probability of being the term for each candidate on the basis of feature values

Methods for term recognition (cont'd)

Feature: a mapping of a candidate into a certain number

Method: a sequence of actions to obtain a ranked list of candidates for a given document collection, which involves calculating one or several features

In the paper, “*feature*” and “*method*” are used interchangeably

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation:

I. Methods based on Statistics of Term Occurrences:

a) TF: term frequency in whole document collection

b) TF-IDF:

$$TF \cdot IDF(t) = TF(t) \cdot \log \frac{1}{DF(t)} \qquad (1)$$

DF(*t*): the number of the documents containing the term candidate *t*

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont'd):

I. Methods based on Statistics of Term Occurrences (cont'd):

c) Domain Consensus: recognition of terms uniformly distributed over the whole collection:

$$DC(t) = -\sum_{d \in Docs} \frac{TF_d(t)}{TF(t)} \cdot \log_2 \frac{TF_d(t)}{TF(t)} \qquad (2)$$

d) word association measures (applied only to multiword terms (often, only to two-word terms)): *z* test [39], *t* test [40], χ^2 test, likelihood ratio [41], mutual information (*MI* [42], *MI*^{2}, and *MI*^{3} [43]), lexical cohesion [44], and term cohesion, etc.

- shown [20, 34] to provide no increase in efficiency

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

I. Methods based on Statistics of Term Occurrences (cont'd):

e) C-value:

$$C-Value(t)=\begin{cases}\log_2|t|\times TF(t) & \text{if } S=\{s{:}t\subseteq s\}=\phi;\\ \log_2|t|\times\left(TF(t)-\frac{\sum_{s\in S}TF(s)}{|S|}\right) & \text{otherwise}\end{cases}\qquad (3)$$

|*t*|: the length of the candidate *t* (in words)

TF(*t*): the frequency of *t* in the text collection

S: the set of the candidates that enclose the candidate *t*, i.e., the candidates such that *t* is their substring

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

I. Methods based on Statistics of Term Occurrences (cont'd):

e) C-value (cont'd):

The weight of the candidate is reduced if this candidate is a part of other candidates, since the candidate frequency in this case is added to the frequency of enclosing candidates: e.g. the frequency of the word combination *point arithmetic* is not less than that of the term *floating point arithmetic*, although the former is obviously not a term.

Disadvantage: only for recognition of multi-word terms

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

I. Methods based on Statistics of Term Occurrences (cont’d):

f) generalized C-value [36]:

$$C - Value(t) = \begin{cases} c(t) \times TF(t) & \text{if } S = \{s : t \subseteq s\} = \phi; \\ c(t) \times (TF(t) - \frac{\sum_{s \in S} TF(s)}{|S|}) & \text{otherwise} \end{cases} \qquad (4)$$

where $c(t) = i + \log_2 |t|$ The authors got the best efficiency when *i*=1

g) generalized C-value [35]:

$$C - Value(t) = \begin{cases} \log_2 (|t|+1) \times TF(t) & \text{if } S = \{s : t \subseteq s\} = \phi; \\ \log_2 (|t|+1) \times (TF(t) - \frac{\sum_{s \in S} TF(s)}{|S|}) & \text{otherwise} \end{cases} \qquad (5)$$

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

I. Methods based on Statistics of Term Occurrences (cont'd):

h) Basic [17](used in PostRankDC)(for recognizing multi-word terms of average specificity):

$$Basic(t) = |t| \log TF(t) + \alpha |\{s : t \subseteq s\}| \qquad (6)$$

In contrast to the *C*-value (in which the frequency of a candidate is reduced if it is part of other candidates), in the Basic, the candidates that contain a given candidate increase its feature value, since average-specific terms are often used to form more specific terms

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

II. Methods Based on Contexts of Term Occurrences:

assumption: The contexts of terms and common words are different.

a) NC-value [24]:

Step 1: The best 200 terms recognized using any method (e.g. C-value);

Step 2: weights of context words: $weight(w) = \frac{t(w)}{n}$ (7)

w: the context word (noun, verb, or adjective);

t(*w*): the number of terms, in the context of which the context word occurs (not to be confused with the term frequency);

n: the total number of terms.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

II. Methods Based on Contexts of Term Occurrences (cont'd):

a) NC-value [24] (cont'd):

Step 3:

$$NC(t) = 0.8 \times C - Value(t) + \sum_{w \in C_t} f_t(w) weight(w) \quad (8)$$

C_t : a set of the words occurring in the context of the candidate *t*,

w: a word from C_t,

$f_t(w)$: the frequency of the word *w* in the context of the candidate *t*.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

II. Methods Based on Contexts of Term Occurrences (cont’d):

b) DomainCoherence (a modification of the NC-value for recognizing of average-specific terms):

domain model: constraints on context words:

(1) occurrence in at least a quarter of the input document collection;
(2) belonging to nouns, verbs, or adjectives;
(3) semantic relatedness to many specific terms.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

II. Methods Based on Contexts of Term Occurrences (cont'd):

b) DomainCoherence (cont'd):

to calculate semantic relatedness of a candidate word *w* for the domain model:

$$s(w) = \sum_{t \in T} PMI(t, w) = \sum_{t \in T} \log\left(\frac{P(t, w)}{P(t) \times P(w)}\right) \qquad (9)$$

T : the set of the best 200 terms recognized by the Basic,

P(*t*, *w*): the probability that the word *w* occurs in the context of the term *t*,

P(*t*), *P*(*w*): the probabilities of occurrence of the term *t* and the word *w*, respectively.

context: a window of 5 words

probabilities: estimated with term frequency in the input document collection.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

II. Methods Based on Contexts of Term Occurrences (cont'd):

b) DomainCoherence (cont'd):

To find the final value of the DomainCoherence, the PMI metric is also used, which is calculated between each term candidate (*t*) and the word from the domain model (*w*).

during the experimental research, the best results were shown by a *linear combination* of the Basic and DomainCoherence, which was called PostRankDC.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

III. Methods Based on Topic Models:

The majority of features based on topic modeling are modifications of the standard methods that use the probability distribution by topics of words (term candidates) instead of the term frequency. Such methods can be applied only to recognition of one-word and (more rarely) two-word terms.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

III. Methods Based on Topic Models (cont'd):

a) i-SWB [47] (can recognize term of any length):

To calculate the termhood of term candidate, one needs distributions of words over the following topics:

- ϕ^{t}, particular topics of the domain ($1 \leq t \leq T$; the authors set $T = 20$);
- ϕ^{B}, background topic;
- ϕ^{D}, topic specific to the document.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

III. Methods Based on Topic Models (cont'd):

a) i-SWB [47] (cont'd):

Then, the most probable 200 words are recognized for each topic (V_t, V_B, and V_D), and the weight of each candidate c_i, which consists of L_i words ($w_{i1}\ w_{i2}\ \dots\ w_{iL_i}$), is taken as a sum of maximum probabilities of the words constituting this candidate (from the distributions found):

$$weight(c_i) = \log(TF(c_i)) \cdot \sum_{1 \le j \le L_i,\, w_j \in \cup\{V_t\}_{t \in T \cup \{B,D\}}} \phi_{w_j}^{mt_{w_j}} \qquad (10)$$

$$mt_{w_j} = \arg\max_{t \in T \cup \{B,D\}} (\phi_{w_j}^{t})$$

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

IV. Methods based on external (reference) corpora (collection of texts of general domain or texts that do not belong to any domain):

1) TF-RIDF: when calculating the number of documents in which the term occurs (IDF (RIDF)), the external corpus is used instead of the domain collection.

2) Domain Pertinence:

$$DP(t) = \frac{TF_{t\arg et}(t)}{TF_{reference}(t)} , \qquad (11)$$

$TF_{target}(t)$: the frequency of the candidate t in the input domain-specific document collection;

$TF_{reference}(t)$: the frequency of t in the general corpus.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

IV. Methods based on external (reference) corpora (cont'd):

3) Domain Relevance:

$$DR(t) = \frac{TF_{t\arg et}(t)}{TF_{t\arg et}(t) + TF_{reference}(t)} \quad , \qquad (12)$$

4) Weirdness (additionally takes into account the size of the collection):

$$W(t) = \frac{TF_{target}(t) \times |Corpus_{reference}|}{TF_{reference}(t) \times |Corpus_{target}|} \quad , \qquad (13)$$

5) Relevance:

$$Rel(t) = 1 - \frac{1}{\log_2\left(2 + \frac{TF_{target}(t) \times DF_{target}(t)}{TF_{reference}(t)}\right)} \quad . \qquad (14)$$

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

IV. Methods based on external (reference) corpora (cont'd):

6) Domain Specificity:

$$DS(t) = \frac{\sum_{w_i \in t} \log \frac{P_d(w_i)}{P_c(w_i)}}{|t|} , \qquad (15)$$

$|t|$: the number of words in the candidate t;

w_i: part of the candidate t;

$P_d(w_i)$: the probability that the word w_i occurs in the domain-specific text collection;

$P_c(w_i)$: the probability that the word occurs in the external corpus.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

V. Methods based on Retrieval Engines:

1) filtration of two-word terms:

submitting requests to retrieval engines: "*A*" (the term itself), "*A* is a term," "*A* is a concept," "A_1," "A_2," and "A_1 AND A_2," where A_1 and A_2 are the words of which the term *A* is composed.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

V. Methods based on Retrieval Engines (cont'd):

1) filtration of two-word terms (cont'd):

For the term to pass the filtration, at least one of the following conditions must be fulfilled:

a) $\frac{hits(\text{"}A \text{ is a term"})}{hits(A)} > C_1$, b) $\frac{hits(\text{"}A \text{ is a concept"})}{hits(A)} > C_2$,

c) $\frac{hits(\text{"}A_1 \text{ AND } A_2\text{"})}{\min(hits(A_1), hits(A_2))} > C_3$,

d) *A* is described by a Wikipedia article

hits(*A*): the number of pages returned by the retrieval engine on the request *A*

C_1, C_2, C_3 ∈ [0, 1] : parameters

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VI. Methods based on Ontologies:

Ontologies: used more rarely than other external resources:

a) general ontologies insufficiently cover domains and include only the most general terms;

b) domain-specific ontologies are available only for a few domains, and the format and structure of such ontologies often depend on a particular domain.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VI. Methods based on Ontologies (cont'd):

1) Dobrov and Loukachevitch used a thesaurus of information retrieval.

Features can only be used for two-word terms:

a) SynTerm: =1 iff, for each word constituting the term, there is a synonym in the thesaurus;

b) Completeness: sums up the synonyms and relations for descriptors, which, in turn, are also found in the thesaurus for individual words of the term

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

VII. Methods based on Wikipedia:

1) In [16], terms are recognized only in Wikipedia, rather than in domain-specific text collections

a) manually select several concepts (Wikipedia articles) as positive examples of domain-specific terms.

b) construct a *weighted graph*, in which nodes are Wikipedia articles and categories, while edges are hyperlinks between them.

c) using manually selected concepts, a *random walk algorithm* is applied to the graph. The weight assigned by the algorithm to each concept is taken as an estimate that the corresponding concept is expressed by a domain-specific term.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

VII. Methods based on Wikipedia (cont'd):

2) method proposed by Vivaldi et al.[59,60]:

In a domain-specific text collection, term candidates are recognized and, then, are estimated by applying path searching algorithms to the graph of Wikipedia categories.

Need to specify *domain borders* (one or several Wikipedia categories that precisely describe a given domain)

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

2) method proposed by Vivaldi et al.[59,60] (cont'd):

Estimating term candidates:

a) For each candidate => all its concepts (Wikipedia articles with the same title) (generally, there can be several articles for one candidate, which is due to lexical polysemy);

b) for each article => all categories to which this article belongs;

c) from all estimates obtained, the best one is selected for each term candidate

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

2) method proposed by Vivaldi et al. [59,60] (cont'd):

Estimating term candidates (cont'd):

d) for each category, the graph of categories is recursively traversed (following only the links to the top-level category) until the specified domain border or the topmost-level category is reached;

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

2) method proposed by Vivaldi et al. [59,60] (cont'd):

Estimating term candidates (cont'd):

e) the properties of the paths found are used to estimate term candidates based on one of the following criteria:

criterion 1. the number of paths:

$$NC(t) = \frac{NP_{domain}(t)}{NP_{total}(t)} \qquad (16)$$

$NP_{domain}(t)$: the number of paths from the categories of the candidate to the domain border;

$NP_{total}(t)$: the number of paths from the categories of the candidate to the top-level category

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

2) method proposed by Vivaldi et al. [59,60] (cont'd):

Estimating term candidates (cont'd):

e) criterion 2. length of paths:

$$LC(t) = \frac{LP_{total}(t) - LP_{domain}(t)}{LP_{total}(t)} \quad , \qquad (17)$$

LP$_{domain}$(*t*): the (total) length of paths from the categories of the candidate to the domain border;

LP$_{total}$(*t*): the (total) length of paths from the categories of the candidate to the top-level category.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

2) method proposed by Vivaldi et al. [59,60] (cont'd):

Estimating term candidates (cont'd):

e) criterion 3: Average length of paths (LMC):

$$LMC(t) = \frac{ALP_{total}(t) - ALP_{domain}(t)}{ALP_{total}(t)}, \qquad (18)$$

NC criterion: the maximum efficiency.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

2. Feature computation (cont'd):

VII. Methods based on Wikipedia (cont'd):

3) LinkProbability (useful for filtering words and phrases belonging to the general vocabulary):

$$LinkProb_T(t) = \begin{cases} 0, & \text{if } t \text{ is not contained in Wikipedia or } \frac{H(t)}{W(t)} < T, \\ \frac{H(t)}{W(t)} & \text{otherwise,} \end{cases} \qquad (19)$$

H(*t*) shows how often the candidate *t* occurs in Wikipedia articles in the form of a hyperlink caption;

W(*t*) shows how often *t* occurs in Wikipedia in total;

T : parameter

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

4) KeyConceptRelatedness:

Step 1: Find key concepts in a given domain-specific document collection:

(a) recognize d key concepts in each document of the collection (d = 3);

(b) select N key concepts with the highest frequency (N = 200).

Step 2: For a given term candidate, find all Wikipedia concepts such that their captions coincide with the term candidate.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

4) KeyConceptRelatedness (cont'd):

Step 3: For each concept found for the term candidate, calculate its semantic relatedness to the key concepts using the weighted *kNN* method adapted for the case of positive examples only:

$$sim_k(c, C_N) = \frac{1}{k}\sum_{i=1}^{k} sim(c, c_i) \quad , \qquad (20)$$

c : the concept of the term; C_N : the set of key concepts ranked in the descending order of semantic relatedness to *c*; *sim*(*c*, c_i): the semantic relatedness function found by the Dice formula, where the articles connected by at least one hyperlink are regarded as neighbors; *k*: the number of the nearest concepts

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

2. Feature computation (cont’d):

VII. Methods based on Wikipedia (cont'd):

4) KeyConceptRelatedness (cont'd):

Step 4: Select the maximum value over all concepts of the term candidate.

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

3. Methods of Feature-Based Inference:

1) Linear combination of features with manually fitted coefficients (generally, equal)

2) Voting algorithm: $$V(t) = \sum_{i=1}^{n} \frac{1}{rank(F_i(t))}, \qquad (21)$$

n: the number of features;

rank(F_i(*t*)): the ordinal number of the candidate *t* among all candidates ranked by the value of the feature F_i.

3) Supervised machine learning (using manually labeled data): Ada Boost [62], logistic regression [33, 53, 63], Random forest [33], and Gradient Boosting [34]

Methods for term recognition (cont'd)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont'd):

3. Methods of Feature-Based Inference (cont'd):

4) Fault-Tolerant Learning (Supervised machine learning (do not use labeled data), a combination of bootstrapping and co-training algorithms):

a) Two sets of features: standard TF-IDF; features based on word delimiters => two lists of candidates consisting of the same elements

b) For each list => the best 500 and the worst 500 candidates as positive and negative examples.

c) training SVMs with five features (candidate frequency, parts of speech for words of the candidate, word delimiters from occurrence contexts of the candidate, the first word of the candidate and the last word of the candidate).

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

3. Methods of Feature-Based Inference (cont'd):

4) Fault-Tolerant Learning (cont'd):

d) applying trained SVMs to all term candidates (1 iteration)

e) repeat step b), c) and d)

Using verification of training sets to avoid degradation of the process:

When different labels (term and non-term) are assigned by two classifiers to the same candidate, this candidate is eliminated from the training set.

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

3. Methods of Feature-Based Inference (cont'd):

5) method proposed in [61]:

modified Basic => 100 best candidates as positive examples

=> training a model of the positive-unlabeled (PU) learning algorithm

=> probabilistic classification of each term candidate

=> recognized candidate filtration according presence in Wikipedia

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

3. Methods of Feature-Based Inference (cont'd):

6) method proposed in [14] (classifies each occurrence of the term candidate individually):

positive examples: words or word combinations that immediately precede a reference to an illustration in the text of a patent;

negative examples: words or word combinations that occur in patents only once or are either citations or units of measurement

Methods for term recognition (cont’d)

- General scheme for the scenario that does not distinguish between occurrences of one term (cont’d):

3. Methods of Feature-Based Inference (cont'd):

6) method proposed in [14] (cont'd):

=> supervised learning (logistic regression and conditional random fields) with 74 features (e.g. parts of speech, contexts and statistics of occurrences)

disadvantage: impossible to transfer to other domain and other languages because of the heuristics used for recognizing positive examples

Efficiency evaluation method

- Two principal approaches for estimating term recognition methods:

  1) manual evaluation by experts in the corresponding domain

  advantage: most accurate evaluation

  2) using preset list of reference terms (gold standard)

  advantage: reproducibility of results, tuning of parameters and comparison between different methods on one dataset

Efficiency evaluation method (cont'd)

- the 2nd approach for estimating term recognition methods:

evaluation techniques of the second approach based on the way of obtaining the list of reference terms:

a) manual labeling of all documents (most accurate but most time-consuming)

b) manual labeling of a small part of documents

c) adaptation of available resources to the term recognition problem, e.g. manually-constructed thesauri or vocabularies, key phrases consisting of key words of papers in one scientific field and terms in subject indexes of books

Efficiency evaluation method (cont'd)

- Efficiency evaluation metrics:
- For the scenario that does not distinguish between occurrences of one term:

1) Precision (or precision at the level N):

$$P(N) = \frac{|\mathrm{Re}\, ference \cap \mathrm{Re}\, trived[1:N]|}{N} \quad , \qquad (22)$$

2) Recall (evaluated implicitly, depending on *P*(*N*) and *N*):

$$R(N) = \frac{|\mathrm{Re}\, ference \cap \mathrm{Re}\, trived[1:N]|}{|\mathrm{Re}\, ference|} \quad , \qquad (23)$$

3) Average precision (most popular):

$$AvP(N) = \sum_{i=1}^{N} P(i)(R(i) - R(i-1)) \quad , \qquad (24)$$

Efficiency evaluation method (cont'd)

- Datasets:
- open datasets:
  - GENIA: 2000 labeled documents on biomedicine; probably the most popular dataset for testing efficiency
  - FAO: 780 manually-labeled reports of the Food and Agriculture Organization (for each report, two terms were recognized)
  - Krapivin: 2304 papers on informatics; as a reference set of terms, key words selected by the authors of the papers are used
  - Patents: 16 manually-labeled patents on electrical engineering
  - Board games: 1300 descriptions and reviews of board games, in which 35 documents (out of 1300) are labeled manually

Experimental comparisons

- Experiments carried out in [20] show that, despite the fact that word association measures are based on the theory of mathematical statistics, their efficiency is comparable to that of the standard term frequency.

- Z. Zhang et al. [21] experimentally compared the following methods, which are capable of recognizing both one-word and multi-word terms: TF-IDF [22], Weirdness [23], C-value [24], Glossex [25], and TermExtractor [26].

  => the results differed depending on the datasets used. The survey also demonstrates the superiority of the *voting algorithm* as a method that combines several features.

Experimental comparisons (cont'd)

- P. Braslavskii and E. Sokolov [27] compared four methods for recognition of two-word terms: term frequency, *t* test, χ^2 test, and likelihood ratio.

  => the first two methods showed the best (comparable) results.

- The same authors [28] also compared five methods for recognizing terms of arbitrary structure: MaxLen [29], *C*-value [24], *k*-factor [30], Window [31] and AOT [32].

  => The methods generally yield similar results, however the *C*-value and the *k*-factor have the highest efficiency, while the AOT has the lowest efficiency.

Experimental comparisons (cont'd)

- In [33], two methods based on combination of several features are compared—voting algorithm and method based on supervised machine learning (logistic regression and Random forest)

  => the second method outperforms the first one.

Experimental comparisons (cont'd)

- M. Nokel and N. Loukachevitch [34] compared methods for recognizing one-word and two-word terms for the problem of thesaurus construction and information retrieval.

  => (1) the best features for recognition of one-word terms are based on *topic models*;

  (2) in all cases, the *combination* of several features yields a considerable increase in efficiency as compared to the use of individual features;

  (3) features based on the *external corpus* offer the most significant increase in efficiency for recognition of two-word terms;

  (4) word association measures provide no increase in efficiency.

Potential development prospects

- developing:

  1) datasets,

  2) experimental research methodologies,

  3) methods for adapting present algorithms to other domains and applications

Reference

- N. A. Astrakhantsev, D. G. Fedorenko & D. Yu. Turdakov. "Methods for automatic term recognition in domain-specific text collections: A survey." *Programming and Computer Software* 41, no. 6 (2015): 336-349.

Q & A

Thank you for listening!