Research Paper (September 2005)

Recognizing Swedish acronyms and their definitions in biomedical literature

Department of Swedish language
GÖTEBORG UNIVERSITY
Dana Dannélls
cl2ddoyt@cling.gu.se

Abstract. This paper addresses the task of recognizing acronyms and acronym definitions in Swedish medical texts (the task can be generalized to other types of text). There are many approaches for identifying acronym-definition pairs in biomedical texts, however none of those addresses the variation and complexity exhibited in the Swedish language. This project investigates some of the previous approaches and present the challenges involved in recognizing Swedish acronym-definition pairs. For processing and extracting acronym-meaning pairs a system called SARP (Swedish Acronym Recognition Program) is introduced. SARP was designed to deal with the task of finding acronym-definition pairs in Swedish texts. The program was evaluated on a hand tagged acronym corpus and performance was measured using standard measures: recall, precision and f-score.

Keywords: Medical text, Swedish text, Data Mining, NLP, acronym recognition

1. Introduction

Dictionaries are essential tools for understanding a language and are frequently used in many technical fields such as biomedicine where the vocabulary is quickly expanding. One known phenomenon in biomedical literature is the growth of new *acronyms*.

An acronym is a shortening for a certain definition, it is defined according to encyclopedia Britannica as "A word formed from the initial letter or letters of each of the successive parts or major parts of a compound term". An acronym can be of any length and may be realized in different surface forms, especially in biomedical texts.

An example of an acronym-definition pair (the acronym together with its related meaning) is: <" **Jama**", "**J**ournal of the **A**merican **M**edical **A**ssociation">.

Many acronym-definition pairs in biomedical texts follow a simple pattern, in which the initial letter of each word in the definition corresponds to one letter in the acronym (as in the previous mentioned example “**Jama”**).

Another common pattern which is normally found among the acronym-definition pairs is the match of suffix and prefixes, such as: **<” vCJD**”, “**v**ariant **CJD ”>** and **<” APOE**”, “**Apo**lipoprotein **E**”**>**. However the variety of the acronym-definition pairs is large and many patterns differ from the examples mentioned above. There exist acronym-definition pairs which include Arabic numbers: **<” MK-4**”, ”**m**ena**k**vinon**-4**”**>**, punctuation: **<” E.coli**”, “**E**scheria **coli**“**>** and other alphanumeric that will be exemplified in sections 3 and 4.

The task of recognizing acronyms, along with their definitions can be defined as a problem of data mining i.e. the process of discovering patterns in data, Yeates (1999), Pustejovsky et al. (2001) [1]. Data is mined to anticipate behavior patterns and trends in natural language texts. It can be used to identify and locate distinguishing characteristics (in this case, relational characteristics that comprise the acronym-pairs). Once such characteristics are found, they can be put to work identifying new targets. In data mining the data is stored electronically and the search is automated.

This project deals with the task of automatically detecting and extracting acronym-definition pairs. This task relies on the approach that acronym-definition pairs follow a set of patterns and other regularities that can be usefully used for acronym identification.

One motivation for this work is that explicit recognition of acronym-definition pairs will enable different algorithms to be tested and compared on the same set of problems for the purpose of automatically:

- creating a self updated dictionary of acronyms.
- annotating acronym-meaning relations that will be used as a tool for co-reference resolution techniques.

This recognition task deals with acronyms found alongside a definition, however it is not always the case that an acronym appears with its corresponding definition, which is also a good motivation to why an acronym dictionary is needed.

Section 2 gives an overview of previous work and discusses some aspects of how this work is related to the specific task of finding Swedish acronym pairs in Swedish medical text. Section 3 outlines the design of the SARP system and section 4 presents the result for SARP performance. In Section 5 some conclusions are drawn and section 6 discusses the future work needed for creating an accurate and stable dictionary that consists of acronym-definition pairs.

[1] http://www.biomedcentral.com/1471-2105/6/103

2. Background

The task of automatically extracting valid acronym-definition pairs from biomedical literature has been studied, almost exclusively for English, over the past few decades using technologies from Natural Language Processing (NLP). There are many electronic acronym dictionaries available online allowing acronym search and submissions of new acronym pairs into the system. These dictionaries often suffer from incompleteness, a problem that increases the need of creating automatic acronym identification systems that will automatically be updated and will not be domain specific. This section presents a few approaches to the acronym identification task and discusses how some of these strategies could be applied to the Swedish acronym recognition problem.

The Longest Common Subsequence (LCS) algorithm Cormen, Leiserson et al. (1990), finds the longest subsequence that two sequences have in common, regardless of the length and the number of intermittent non- matching symbols. For example, the sequences “abcD” and “aXbc” have a length three sequence “abc” as their longest common subsequence.
The algorithm achieved good result and was therefore tested on some of the Swedish acronym pairs. In order to test the algorithm it was implemented in java. The LCS algorithm can be described as follows: Given a sequence X[0,i] = X_0 X_1 X_2 ...Xi and a sequence Y[0,j] = Y_1 Y_2 Y_3 ...Yj.
Find a sequence L[i,j] that denotes the length of a longest string that is a subsequence of both X and Y.
The algorithm determines whether Xi = Yj at any position L[i,j].
If this is true it takes Xi as the next character of the subsequence, moving next to L[i-1,j-1]. If Xi != Yj then it moves to the larger of L[i,j-1] and L[i-1,j].
The search ends when the algorithm reaches a boundary cell
with i = -1 or j = -1.

The results of testing some of the acronym-definition pairs shows that the algorithm attempts to find as many matches as possible which leads to confusion and results in incorrect match, for example the algorithm matched the underlined letters: <” **idf**”, “International D**i**abetes Fe**d**eration”>
instead of: <” **idf**”, “**I**nternational **D**iabetes **F**ederation”>.

An observation concerning lower and upper case letters match, such as in the acronym-definition pair <”**MR**”, “**m**agnet**r**esonanstomografi”> are not handled by the algorithm i.e. no match is found for the characters ‘M’ and ’R’.

Taghva and Gilbreth (1999), present the AFP (*Acronyms Finding Program*). The program seeks for acronym candidates (words with three to ten characters in length), which appear as upper case words. The search space starts by testing whether the acronym candidate exists in a list of rejected words (words like

“USA”, “47th”) , then the previous and following text is searched for the acronym definition, the search space (length in words) of the definition is limited to 2n words, where n is the number of characters in the acronym candidate.
They calculate a heuristic score for each competing definition by classifying words into: stop words (“the”, “of”, “and”), hyphenated words (for instance <” **IGF-1**”, “**i**nsulin-like **g**rowth **f**actor-**1**“> the word “like” should be ignored), normal words (words that don’t fall into any of the above categories) and the acronyms themselves (since an acronym can sometimes be a part of the definition).
The initial letters of the words found match against the letters of the acronym itself, using the LCS algorithm.

One of the major drawbacks of the AFP is that it seeks only for the acronym candidates which appear in uppercase letters. Acronym candidates such as <” **als**”, “**a**myotrofisk **l**ateral **s**kleros”> are not considered by the system. The program also fails to match such acronyms that consist of two characters such as <” **PR**”, “**p**er **r**ectum”>, a structure which is rather common among the Swedish acronym pairs.

Similar approach to Taghva and Gilbreth was presented by Yeates (1999). Yeates introduce an alternative program to the AFP, called TLA (*Three Letters Acronyms*), using more complex methods to match characters of the acronym candidate with letters in the definition words. Yeates present some general heuristics to find acronym candidates and notes the challenge of finding optimal weights for each heuristic. The TLA program performs better then the AFP but has some of the above mentioned limitations concerning the Swedish acronym identification problems.

Another approach recognizes that the alignment between an acronym and its definition often follows a set of patterns, Byrd and Park (2001), Larkey et al.(2000). Pattern-based recognition gives strong constraints to the acronym respectively the definition recognition and ensures reasonable precision but it might limit recall. To avoid low recall, Nadeau and Turney (2005) use weak constraints to reduce the search space of the candidate acronyms and the candidate definitions.

Schwartz and Hearst (2003) present a simple algorithm for extracting abbreviations from biomedical text. The algorithm tries to extract acronyms candidates assuming that either the acronym or the definition occurs in parentheses, giving some restriction for the candidate definition such as length and capital letter initialization. When an acronym candidate is found the algorithm scans the words in the right and left side of the found acronym and tries to match the shortest definition that matches the letters in the acronym. Their approach is based on previous work (Pustejovsky et al., 2001),

achieving high recall and precision. The algorithm seeks only for acronyms or definitions that are surrounded by parentheses, hence cases where neither the acronym nor the definition appears within parentheses are not found.
It should emphasized that the common characteristic of most previous approaches in the surveyed literature, is the use of parentheses as indication for the acronym pairs, Nadeau and Turney (2005), table 1, pp.4-5.

Considering the large variety in the Swedish acronym-definition pairs it is practical to use pattern-matching techniques. Such techniques will enable to extract relevant information of which a suitable set of schema will give a representation valid to present the different acronym pairs (section 6). Finding useful patterns that describe acronym-definition pairs is crucial for the recognition task and may lead to some advantages that will allow making non-trivial prediction on new data. Next section describes the design and implementation of SARP. The program uses similar algorithm described by Schwartz and Turney, with some contributions.

3. Methods and implementation

Some choices were made when designing the method for coding acronyms with respect to their definitions. Term formation patterns are used for handling acronym variations. These describe the different acronym forms and the definition that is likely to match them. The pattern-based recognition that is used in SARP relies on some predefined strong constrains that limit the number of the acronym candidates.

3.1 Acronym Candidates

A valid acronym candidate is a string of alphabetic, numeric and special characters such as '-' and '/'. It is found if the string satisfies the conditions (i) and (ii) and at least one of the three remaining conditions (iii) or (iv) or (v):

(i) The string length is longer than 2 characters.
(ii) The string is not in the list of rejected words, e.g. 'USA', 'FN', 'UN'.
(iii) The string contains at least one capital letter.
(iv) The string first or last character is lower case letter or numeric.
(v) If parentheses are found and the string doesn't contain a capital letter the length of the string is no longer than 3 characters.

3.2 Definition Candidates

When an acronym is found SARP searches the words surrounding the acronym for a substring of a valid definition candidate that satisfies all of the following conditions:

(i) At least one letter of the words in the substring matches the letter in the acronym.

(ii) The substring doesn’t contain a colon, semi-colon, question mark or exclamation mark.

(iii) The maximum length of the substring is the length of the acronym length +5 or the acronym length *2 ,Park and Byrd (2001).

(iv) The substring doesn’t contain only upper case letters.

(v) The length of the substring must be greater than the length of the acronym candidate.

The above mentioned constraint allows most of the valid acronym entities to be successfully recognized along with their definition. Some examples of particular structures that are allowed according to these constraints are: <” **mRNA**”, “budbärar**RNA>**”,<” **aeds**”, “**a**topiskt **e**ksem/**d**ermatit **s**yndrom”>, <” **H-II**”, “**H**emocult **II**”>, <” **IVF/ET**”, “**i**n **v**itro-**f**ertilisering/**e**mbryo **t**ransfer”>.

3.3 Matching Acronyms and Definitions

The process of extracting acronyms and their definition from a raw text, according to the above mentioned constraints, is divided into two steps:

1. Parentheses matching.

In practice, most of the acronym-definition pairs come inside parentheses (Schwartz and Hearst) and can correspond to two different patterns:

a) definition (acronym).
b) acronym (definition).

SARP tries to extract all the acronyms or the definitions which correspond to one of these two patterns. This strategy has proven to be very efficient and suitable for this task, because as seen in medical texts an acronym or a definition often occurs within parentheses, it is therefore suitable to search for these patterns first.

2. Non parentheses matching.

SARP seeks for acronym candidates that follow the above mentioned constraints and that are not found within parentheses.
Once an acronym candidate is found the algorithm scan the previous and following context where the acronym was found, trying to match a substring that may be the definition candidate for the found acronym. The search space for the correct substring of a definition candidate is limited to 4 words * the number of letters in the acronym candidate. To illustrate this we assume that the found acronym consists of two characters and is found at the end of the sentence of 16 words long. It is likely that the definition will be found among the last 8 words rather then among the first 8. Assuming that the distance between the acronym and the definition may be up to two words and that the definition may include prepositions, determiners or other words that do not correspond to any letter in the acronym.

When an acronym and a substring for the definition are found, the SARP tries to identify the correct substring of the definition by reducing the first substring. It tries to match identical characters between the acronym and the definition starting from the end of both the acronym and the definition strings. SARP succeeds in finding a correct pair if it satisfies the two following conditions:
(i) at least one character in the acronym matches with a character in the definition;
(ii) the match of the character at the beginning of the acronym string matches a character in the initial position of the leftmost word in the definition substring, ignoring upper/lower case letters.
In this step, acronym-definition pairs are eliminated if there is no letter match is found. For instance: if the string for the found acronym candidate is "FG" and the string for the found definition candidate is "peritoneal dials University Utrecht", they will not be considered as a valid pair.

The found acronym-definition pair candidate is then added to a set of found candidates and compared against a list of correct acronym pairs in order to determine whether the found acronym pair was correctly identified.

4. Evaluation and result

SARP was evaluated on untagged instances of the MEDLEX corpus[2], Kokkinakis (2004). The instances comprise a set of 861 acronym-definition pairs.
The following measurements are used to calculate the acronym-expansion matching performance:

$$\text{precision} = \frac{\text{number of correct acronym-definition pairs returned}}{\text{total number of acronym-definition pairs returned}}$$

$$\text{recall} = \frac{\text{number of correct acronym-definition pairs returned}}{\text{total number of correct acronym-definition pairs in file}}$$

$$\text{F-score} = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}$$

To calculate the total number of the correct acronym-definition pairs in the input file, the same corpus file was hand tagged. Using a separate method (AcronymLexicon.java), a list of all the tagged acronym-definition pairs was extracted. The same list, from the tagged file, was used to count true positive candidates that were found in the untagged file.

[2] http://demo.spraakdata.gu.se/svedk/pbl/MEDLEX_work2004.pdf

The results are summarized in table 1.

	Total in file	Found by SARP	incorrectly identified	Precision	Recall	F-score
unique	725	548	45	92 %	69 %	79 %
Not unique	861	671	52	92 %	72 %	81 %

Table 1: evaluation results of the acronym definition pairs retrieved by SARP.

A closer look at the incorrect acronym pairs that were found shows that SARP failed to match an acronym with its definition when:
(i) words that appear in the definition string don’t have a corresponding letter in the acronym string, for example:
(1) “…<def id="2"> National Cholesterol Education Program Adult Treatment Panel III </def> (<acr link="2"> ATPIII </acr>) and European..”
The found pair was: <” **ATPIII**”, “**A**dult **T**reatment **P**anel **III**”**>.**
(2) “… vid <def> Institutionen för fysiologi och farmakologi </def> (<acr> FYFA </acr>) , Karolinska Institutet…”
The found pair was: <” **FYFA**”, “**fy**siologi och **f**arm**a**kologi”>.
(ii) letters in the acronym string don’t have a corresponding word in the definition string, for example:
(1) “.. propylene <def>glycol alginate lösning</def> (<acr>PGA</acr>) ..”
The found pair was:<” **PGA**”, “**p**ropylene **g**lycol **a**lginate lösning”>.
(iii) words that appear in the definition string don’t match the letters in the acronym string, for example:
(1)“… vid VT eller <def id="2">kammarflimmer</def> (<acr link="2">VF</acr>)…”
The found pair was: <” **VF**”, “**V**T eller kammar**f**limmer”>.
(2) “…<acr link="6">VOC</acr> = <def id="6">organiskt hjärtfel</def> (vitium organicum cordis) .”
The found pair was: <” **VOC**”, “**v**itium **o**rganicum **c**ordis”>.

Some other cases where SARP failed to make a correct match are:
(1) “..concentration of <def>IL-6 in amniotic fluid</def> , <acr>IAI</acr> ..**“**
The found pair was: <” **IAI**”, “**i**n **a**mniotic flu**i**d”>.
(2) “.. och <def id="2">Svenska Sällskapet för Pharmaceutical Medicine</def> , <acr link="2">SSPM</acr> **. ”**
The found pair was: <” **SSPM**”, “**S**äll**s**kapet för **P**harmaceutical **M**edicine”>.
(3) “...livskvalitet-enkät - " <def>the Multi-Clinic Smell and Taste Questionnaire</def> " (<acr>MCSTQ</acr>) .“
The found pair was: <” **MCSTQ**”, ” **M**ulti-**C**linic **S**mell and **T**aste **Q**uestionnaire ”>.
(4) “ ..ofta genom <def>New York Heart Associations</def> klassificering (<acr>NYHA</acr>)..”
The found pair was: <” **NYHA**”, “**N**ew **Y**ork **H**eart **A**ssociations klassificering”>.

One of the drawbacks of the SARP is that it tries to match characters starting at the end of both the acronym and the definition strings using back forward search algorithm. To increase recall it is necessary to combine forward search algorithm that tries to match characters starting at leftmost side of the strings.

It appears that acronym-definition pairs are also marked with special symbols such as '=' '[..']' that imply their existence, for example here is a line taken from the corpus "….valsartan (VALUE [Valsartan Antihypertensive Long-term Use Evaluation….])…."
The algorithm found the following acronym-definition pairs:
<" **valsartan**", **"V**alsartan **A**ntihypertensive **L**ong-term Use Evaluation">,
and <" **VALUE**", "**V**alsartan **A**ntihypertensive **L**ong-term **U**se **E**valuation">.
The second mentioned is the correct acronym pair.
Including such text markers in the code might exclude incorrect extraction of acronym candidates, such as the first mentioned example and thereby increase precision.

Some examples of the acronym-definition pairs that were not found:
(1)**"** ..<def>Usher typ III</def> (<acr>USH3</acr>)…"
Failed to find: <" **USH3**", " **Ush**er typ III ">
(2) **"..** <def id="3">39-item Parkinson's disease Questionnaire</def> (<acr link="3">PDQ 39</acr>) .."
Failed to find: <" **PDQ 39**", **"** **39**-item **P**arkinson's **d**isease **Q**uestionnaire">
(3) "..av <def>icke småcellig lungcancer</def> (<acr>NCLC</acr>) …"
Failed to find: <" **NCLC**", "icke småcellig **l**ung**c**ancer ">
(4) ".. hur <def>endoplasmatiskt retikel</def> (<acr>SER</acr> och RER) "
Failed to find: <" **SER**", "**e**ndoplasmatiskt **r**etikel ">.

Another example of an acronym candidate that was not found by SARP is:
" ..de så kallade <def>apolipoproteinerna</def> , <acr>apo</acr> ." It failed to find the acronym-definition pair <**" apo", "apo**lipoproteinerna"> .
This is very difficult case because if the system will allow three letter acronyms which consist of only lowercase letters the number of found false candidates will increase drastically. There are many prepositions, verbs and determinates that correspond to this structure and in most cases it is easy to mach their letters with one of the letters in the surrounding words. This kind of acronym is therefore very difficult to detect.

Example of acronyms that correctly identified by the SARP:
(1) "…icke-motorisk <def id="2">test för visuell perceptionsstörning</def> (<acr link="2">TVSPR</acr>)…"
Found: <" **TVSPR**", "**t**est för **v**isuell **p**erceptionsstörning">
(2) "…läkemedelsmyndighet <acr>FDA</acr> (<def>Food & Drug Adminstration</def>) försökt att …"

Found: <″ **FDA″, “F**ood & **D**rug **A**dminstration″>
(3) “…namnet <def>Health Behaviour in School-aged Children</def> (
<acr>HBSC</acr>)...″
Found: <″ **HBSC**″, “**H**ealth **B**ehaviour in **S**chool-aged **C**hildren″>
(4) ″…att en <def>acetylgrupp</def> , <acr>AC</acr> , fästs på det område
av…″
Found: <″ **AC**″, ″ **ac**etylgrupp ″>.

5. Conclusions

The variety of acronym pairs is large and involves different structures which are hard to detect, for example <″ **VF**″, “kammar**f**limmer″>, <″**CT**″, “dator**t**omografi″>, where the acronym is in English while the extension is written in Swedish. In the second mentioned example the occurrence of character **d** instead of **c**. This structure is almost impossible to find in a text if straightforward dictionary/database lookup is not used. It becomes more difficult because there are also counter examples in the Swedish text where both the acronym and the definition are in English, such as <″ **CT**″, ″**c**omputed **t**omography″>.

The system presented in this paper, SARP, relies on already existing acronym pairs which are seen in different Swedish texts. SARP, implemented in Java[3], was designed to deal with the Swedish acronym recognition task. It utilizes predefined strong constrains to find and extract acronym-definition pairs with different patterns.
The use of manually defined constraints that limit the number of the acronym candidates has proven to be a suitable strategy that can be further improved. As suggested in section 4, it may be suitable to combine other algorithm approaches and syntactic structure information with the exiting code, to increase precision and recall.

The result presented for SARP performance is not as good as presented in earlier work that is mostly concerned with the English language, but it should be evaluated on other data before conclusions can be drawn.
It will be interesting to test SARP on a different corpus and observe the result.

6. Future work

The increasing growth of unique acronym pairs gives a need for using machine learning algorithms that will automatically determine whether an acronym pair is valid or not. This is done by using a generated model

[3] The source code is available at http://www.cling.gu.se/~cl2ddoyt/acronyms/acronym.htm

(generated from the training data) which consists of a set of attributes that convey sufficient information about the acronym pairs.
One advantage for applying machine learning techniques is that decisions which are made by a certain learning scheme, based on one set of examples, could later be applied to any given text where unseen acronym pairs will successfully be recognized.
One powerful workbench which allows access to different machine learning algorithms for data mining tasks and is well suited for the acronym recognition task is Weka[4] (distributed under the GNU Public License).
The software is written in java and offers many features, such as comprehensive set of data pre-processing tools, learning algorithms, evaluation methods and an environment for comparing different learning algorithms. A detailed and updated API can be found in the online documentation on the Weka website, it contains an index of all publicly accessible variables and methods in Weka which can easily be accessed and used by any developer.

In order to apply supervised learning approach that will gain knowledge from the data and make relevant predictions from it, it is crucial to choose suitable and meaningful attributes that describe the data.
Observation of the acronym pairs generated by SARP and other observations mentioned in earlier work, Schwartz and Hearst (2003), yielding a few attributes that are extremely relevant and should therefore be considered in the next step, these features are summarized below:

1. Whether the acronym or the definition is between parentheses.
2. The order in which acronym-definition pairs are found (i.e. acronym first, definition second or vice versa).
3. The distance (in words) between the acronym and the definition.
4. The number of letters in acronym that matches the first letter in the definition words (considering both upper and lower case letters).
5. The number of words in the definition that do not participate.
6. The number of words in the definitions that are prepositions or determiners.

Next step before applying machine learning techniques is to compute the above attributes into feature vectors. Each vector will describe each of the acronym-definition pair and will be presented in the input model.
In this model both negative and positive acronym-pairs must be presented.
This will be the next phase in SARP development, namely computing feature vectors for the generated acronym-pairs that were presented in this paper.
It is not clear whether SARP improvement is necessary before applying learning algorithms, but it will be challenging to improve its performance.

[4] http://www.cs.waikato.ac.nz/~ml/weka/index.html

References

Cormen, T.H., Leiserson, C.E. and Rivest, R.L. (1990) Introduction to algorithms. *MIT Press*, Cam bridge, MA.

Taghva, K. and Gilbreth, J. (1999) Recognizing acronyms and their definitions. *Technical Report.* Taghva95-03, ISRI ISRI; November.

Yeates, S. (1999) Automatic extraction of acronyms from text, *Proc New Zealand Computer Sci ence Research Students' Conference, pp. 117–124.* University of Waikato, Hamilton, New Zealand.

Larkey, L., Ogilvie, P., Price, A. and Tamilio, B. (2000) Acrophile: An Automated Acronym Extractor and Server, *In Proceedings of the ACM Digital Libraries conferenc*e, pp. 205-214.

Park, Y., and Byrd, R.J., (2001), Hybrid Text Mining for Finding Abbreviations and Their Definitions, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processin*g, Pittsburgh, PA.

Pustejovsky J. et al. (2001) Automation Extraction of Acronym-MeaningPairs from Medline Databases, *Medinfo*; 10 (Pt 1): 371-375.

Schwartz, A. and Hearst, M. (2003), A simple algorithm for identifying abbreviation definitions in biomedical texts, *In Proceedings of the Pacific Symposium on Biocomputing (PSB).*

Nadeau, David and Turney, Peter (2005) A Supervised Learning Approach to Acronym Identification. In Kégl, Balázs and Lapalme, Guy, Eds. *Proceedings 18th Conference of the Canadian Society for Computational Studies of Intelligence* LNCS 3501, pages pp. 319-329, Victoria, BC, Canada.

Kokkinakis D. (2004). MEDLEX. *Technical Report*. Department of Swedish Language, Språkdata, Sweden.