Technical Reports on Language Technology
Report No 45, 20019
http://www.njas.helsinki.fi/salama

Intelligent search engines[1]

Arvi Hurskainen
Department of World Cultures, Box 59
FIN-00014 University of Helsinki, Finland
arvi.hurskainen@helsinki.fi

Abstract

The traditional way of information retrieval is to type the search string into the search box, and the system finds the matching hits from target text. It has been been possible to use Boolean operators such as AND and OR for combined search. For isolating languages such as English, such a system works quite well. The situation is very diffrent with inflecting languages such as Finnish, which inflects to the right from the stem, or Swahili, which inflects to both directions from the stem. Without proper morphological analysis it is very hard to formulate the search string so that the result is satisfactory. In this report I will describe the latest developments in accurate information retrieval. It is an extension to the Technical Report No. 36 (Hurskainen 2019a) and 44 (Hurskainen 2019b).

Key Words: *information retrieval, morphological analysis, disambiguation.*

1 Introduction

Analysed text corpora are available mostly for research purposes in a few languages. To produce such a corpus is usually a long process and requires often a lot of post-editing. The strength of such corpora is that they make possible a large variety of searches, because the analysis result in the form of tags can be used in formulating search strings.

The access to such advanced search system is normally restricted to science community. The search systems available to everybody are simple string search systems. They are either strict string match systems, or they allow some obvious typing errors, such as the search system of Google. More intelligent search systems have not come to the market.

It is hard to understand why language technology (Note! neural machine translation is not language technology) has not been applied to search systems, although it improves search results tremendously. One reason may be that the dominant world language English is an isolating language with very limited morphology. Therefore, the benefits of language technology in search systems might not be very impressive. The situation is very different with morphologically rich languages, such as Finnish and Swahili.

Technical Report No 44 (Hurskainen 2019b) describes two intelligent search systems. One of them makes use of runtime language analysis. In it, the user may type any inflected form of the search word, and the system finds in analysed target text all

inflected words of that base form. The analyser turns the typed word into base form and attaches the appropriate POS tag to it. The search is then made on the basis of the base form. This form matches in text with all those words, which have this base form.

The other search system is different in that it does not make use of the runtime analysis. Several search alternatives are in this system, because the user formulates the search string himself. The search can be targeted to surface strings or to the base form.

I will describe the latest development of this system in this report.

2 Producing the analysed target text

One reason for the slow spread of intelligent search systems might be that the target text must first be analysed and disambiguated. This should be done to each target text. This requirement seems to limit the use of this system to such texts, which are considerably stable and normal prose text. But does it? Could the target text be analysed and disambiguated in the fly and then the search could be targeted to it?

We should process the target text into such a format, where each surface word is followed by its base form plus the POS tag, as in (1).

(1)
1_§_2 Suomen {Suomi_PROPN} valtiosääntö {valtiosääntö_N} on {olla_V} vahvistettu {vahvistaa_V} tässä {tämä_PRON} perustuslaissa {perustuslaki_N}.

Although this is a simple representation, in order to be able to disambiguate the sentence properly we need a much more detailed description, such as in (2).

(2)
```
"<1_§_2>"
"<*suomen>"
      "suomi"  CAP PROPN SG GEN/ACC
"<valtiosqxqxntqz>"
      "valtiosqxqxntqz"  N SG NOM
      "valtiosqxqxntqz"  N SG ACC-N
"<on>"
      "olla"  V VMOD PRES SG3
"<vahvistettu>"
      "vahvistettu"  A SG NOM
      "vahvistettu"  A SG ACC-N
      "vahvistaa"  V TRV PASS-2PART-NOM
      "vahvistaa"  V TRV PASS/PASS-NEG-PAST
      "vahvistaa"  V TRV PASS A SG NOM
      "vahvistaa"  V TRV PASS A SG ACC-N
"<tqxssqx>"
      "tqxmqx"  PRON DEM SG1 INE
      "tqxssqx"  ADV
"<perustuslaissa>"
      "perustuslaki"  N SG INE
"<.$>"
      "."  **CLB
```

Note that the scands in the analyser have been converted to bigrams. We see that many words have more than one interpretation. Therefore, we must disambiguate it (3).

(3)
```
"<1_§_2>"
"<*suomen>"
      "suomi" CAP PROPN SG GEN/ACC
"<valtiosqxqxntqz>"
      "valtiosqxqxntqz" N SG NOM
"<on>"
      "olla" V VMOD PRES SG3
"<vahvistettu>"
      "vahvistaa" V TRV PASS-2PART-NOM
"<tqxssqx>"
      "tqxmqx" PRON DEM SG1 INE
"<perustuslaissa>"
      "perustuslaki" N SG INE
"<.$>"
      "." **CLB
```

When the text has been disambiguated, it can be reduced to the format, where it has only the surface word and its base form plus its POS tag after it, such as in (4).

(4)
1_§_2 Suomen {Suomi_PROPN} valtiosääntö {valtiosääntö_N} on {olla_V} vahvistettu {vahvistaa_V} tässä {tämä_PRON} perustuslaissa {perustuslaki_N}.

All these phases from raw text into the form above can be produced as a single process. It should be possible to implement the analysis and disambiguation process so that the user retrieves the text from the local machine, processes the text into required form, and then performs search operations on it.

The performance of any analyser on arbitrarily chosen text is seldom perfect, and disambiguation may be even more defective. The situation is different with such fixed texts as *Perustuslaki* and *Raamattu*. Their analysis can be done 100 percen correct, and also their disambiguation near perfect.

On the other hand, do we always need full performance of the analysis system, if we also can perform searches on the basis of surface forms? Therefore, although the analysis leaves some words unanalysed, these words can be retrieved with the surface word search system.

3 Combined search system

When we have the target text with surface and base form representation, we can do many kinds of searches. We can direct the search to any string in the target text. The problem is how to display the hit in the way that it is easy to see it in result text. In the normal procedure, when the hit is found, it will be printed with context, and then all base forms

are removed, so that only the original text remains. The hit is there, but it is in no way marked to help reading.

One solution is to use colour code marking available in *egrep*. When the hit is marked with colour code, it can then be rewritten into various marking systems, depending on the type of hit.

3.1 Base form search

In the search system, where a runtime analyser is used, the typed search word is first analysed and reduced to the combination of base form and its POS tag, such as *{laki_N}*. The user may enter any word form, and it is converted into the base form. This form is then used in actual search. Therefore, this search system can find only base forms.

Also in the other search system, where runtime analysis is omitted, search can be directed to base form. The search key must include at least one such feature, which occurs only in base forms. Such features are {, }, and _. These occur only in base forms.

When the hit is found from the target text, it contains the search word in base form, but also base forms of all other surface words. To add readability of the result, the base forms must be removed. If no further measures are done, also the hit loses its base form.

On order not to lose the hit, the match of the keyword in text mast be marked. This can be done by surroundig the match with colour codes and by converting its boundary marks, { and }, trmporarily into something else, for example into [[and]]. Now the boundary marks of the hit are different from the base form marks of other words. The base forms with boundary marks { and } can be removed, and only the base form of the hit will remain. The temporary marks of the hit word can then be returned back to the original form { and }.

Note that although the entered key word is not full base form, the boundary marks will be put around the full word, whatever it is. For example, if the search word is *laki_,* the result is as in (5).

(5)
1_§_2 Suomen valtiosääntö on vahvistettu tässä perustuslaissa {perustuslaki_N}.
2_§_3 Julkisen vallan käytön tulee perustua lakiin {laki_N}.
2_§_4 Kaikessa julkisessa toiminnassa on noudatettava tarkoin lakia {laki_N}.

The system finds all words, where the last part of the base form is *laki*.

If we mark the beginning of the search word, such as *{laki_*, the system finds only those words (6).

(6)
2_§_3 Julkisen vallan käytön tulee perustua lakiin {laki_N}.
2_§_4 Kaikessa julkisessa toiminnassa on noudatettava tarkoin lakia {laki_N}.
5_§ Suomen kansalaisuus saadaan syntymän ja vanhempien kansalaisuuden perusteella sen mukaan kuin lailla {laki_N} tarkemmin säädetään.

We can also search on the basis of POS tag, such as *A}* or *_A* (7).

(7)
2_§ Valtiovalta Suomessa kuuluu kansalle, jota edustaa valtiopäiville kokoontunut {kokoontunut_A} eduskunta.
2_§_3 Julkisen {julkinen_A} vallan käytön tulee perustua lakiin.
2_§_4 Kaikessa {kaikki_A} julkisessa {julkinen_A} toiminnassa on noudatettava tarkoin lakia.

If we omit a feature unique to the base form, we will also find surface words starting with A.

Adverbs will be found with the key word *_ADV* or *ADV* or *ADV}* (8).

(8)
6_§_3 Lapsia on kohdeltava tasa-arvoisesti {tasa-arvoisesti_ADV} yksilöinä, ja heidän tulee saada vaikuttaa itseään koskeviin asioihin kehitystään vastaavasti {vastaavasti_ADV}.
6_§_4 Sukupuolten tasa-arvoa edistetään yhteiskunnallisessa toiminnassa sekä {sekä_ADV} työelämässä, erityisesti {erityisesti_ADV} palkkauksesta ja muista palvelussuhteen ehdoista määrättäessä, sen mukaan kuin lailla tarkemmin {tarkasti_ADV} säädetään.

Also part of word stem can be in the search word, such as *sti_ADV* (9).

(9)
7_§_2 Ketään ei saa tuomita kuolemaan, kiduttaa eikä muutoinkaan kohdella ihmisarvoa loukkaavasti {loukkaavasti_ADV}.
7_§_3 Henkilökohtaiseen koskemattomuuteen ei saa puuttua eikä vapautta riistää mielivaltaisesti {mielivaltaisesti_ADV} eikä ilman laissa säädettyä perustetta.
9_§ Suomen kansalaisella ja maassa laillisesti {laillisesti_ADV} oleskelevalla ulkomaalaisella on vapaus liikkua maassa ja valita asuinpaikkansa.

3.2 Searching full surface words

It is also possible to search for full surface words. In this method we direct the search to full surface words. Although the search string would match also with the base form, this will be ignored. In the result, only surface words are present and all base forms are deleted. If the key word is *lailla*, the result is as in (10).

(10)
5_§ Suomen kansalaisuus saadaan syntymän ja vanhempien kansalaisuuden perusteella sen mukaan kuin <lailla> tarkemmin säädetään.
6_§_4 Sukupuolten tasa-arvoa edistetään yhteiskunnallisessa toiminnassa sekä työelämässä, erityisesti palkkauksesta ja muista palvelussuhteen ehdoista määrättäessä, sen mukaan kuin <lailla> tarkemmin säädetään.
7_§_6 Vapautensa menettäneen oikeudet turvataan <lailla>.

We see the found full surface words are surrounded with angle brackats < and >.
In searching surface words, capital letters matter. With the search word *Suomen* we get results as in (11).

(11)
1_§_2 <Suomen> valtiosääntö on vahvistettu tässä perustuslaissa.
4_§ <Suomen> alue on jakamaton.
5_§ <Suomen> kansalaisuus saadaan syntymän ja vanhempien kansalaisuuden perusteella sen mukaan kuin lailla tarkemmin säädetään.

3.3 Free search of partial surface words

If only part of surface word is typed in the search key, the system finds those lines, where the search string occurs. The typed string can be in any part of the word. The hit is marked so that the whole word is surrounded with square brackets [and]. With the search word laki we get the result as in (12).

(12)
2_§_4 Kaikessa julkisessa toiminnassa on noudatettava tarkoin [lakia].
18_§_5 Ketään ei saa ilman [lakiin] perustuvaa syytä erottaa työstä.
28_§_5 Jos kansanedustaja olennaisesti ja toistuvasti laiminlyö edustajantoimensa hoitamisen, eduskunta voi hankittuaan asiasta [perustuslakivaliokunnan] kannanoton erottaa hänet edustajantoimesta joko kokonaan tai määräajaksi päätöksellä, jota on kannattanut vähintään kaksi kolmasosaa annetuista äänistä.

In the word *[perustuslakivaliokunnan]*, the hit is in the middle of the word. In the other two examples it is in the beginning of the word.

By typing letters without boundary marks or base word features, the hit is directed to surface words only. Also capital letters matter. With the search word *Suo* we get results as in (13)

(13)
1_§ [Suomi] on täysivaltainen tasavalta.
1_§_2 [Suomen] valtiosääntö on vahvistettu tässä perustuslaissa.
2_§ Valtiovalta [Suomessa] kuuluu kansalle, jota edustaa valtiopäiville kokoontunut eduskunta.

3.4 Controlled search of partial surface words

We can also search surface words in a more controlled way. By placing an asterisk '*' in front or in the end of the string, we force the engine to work in the way, that the typed string before or after the asterisk is the beginning part or end part of the word. The match does not happen, if these conditions are not met. If the search key is a plain string, such as *asia*, we get results as in (14).

(14)
12_§_6 Jokaisella on oikeus saada tieto julkisesta [asiakirjasta] ja tallenteesta.
38_§_3 [Apulaisoikeusasiamiehestä] ja [apulaisoikeusasiamiehen] sijaisesta on soveltuvin osin voimassa, mitä [oikeusasiamiehestä] säädetään.

If we force the match to be the beginning part of the word, we must type *asia** (15).

(15)
31_§_3 Jos kansanedustaja rikkoo tätä vastaan, puhemies voi huomauttaa [asiasta] tai kieltää edustajaa jatkamasta puhetta.
32_§ Kansanedustaja on esteellinen osallistumaan valmisteluun ja päätöksentekoon [asiassa], joka koskee häntä henkilökohtaisesti.
32_§_2 Hän saa kuitenkin osallistua [asiasta] täysistunnossa käytävään keskusteluun.

Now only those lines will be printed, where the keyword *asia* is in the beginning of the word.

If we want that the key word is the end part of the word, we must fromulate the key word with the asterisk in front of the key word, **asia* (16).

(15)
42_§_4 Jos eduskunta ei tyydy puhemiehen menettelyyn, [asia] lähetetään perustuslakivaliokuntaan, jonka tulee viipymättä ratkaista, onko puhemies menetellyt oikein.
58_§_2 Jos presidentti ei päätä asiasta valtioneuvoston ratkaisuehdotuksen mukaisesti, [asia] palautuu valtioneuvoston valmisteltavaksi.

3.5 Boolean operators in search

Search strings can be combined with Boolean operators OR and AND. The operator OR is easier to implement, because it only lists alternative search strings, and any match will be printed. With the search string *laki OR oikeus*, we get the result as in (16).

(16)
2_§_2 Kansanvaltaan sisältyy yksilön <oikeus> osallistua ja vaikuttaa yhteiskunnan ja elinympäristönsä kehittämiseen.
2_§_3 Julkisen vallan käytön tulee perustua [lakiin].
2_§_4 Kaikessa julkisessa toiminnassa on noudatettava tarkoin [lakia].
3_§_3 Tuomiovaltaa käyttävät riippumattomat tuomioistuimet, ylimpinä tuomioistuimina korkein <oikeus> ja korkein [hallinto-oikeus].

We see that if the match is the whole word, the word is surrounded with angle brackets. If the match covers only part of the word, the whole word is surrounded with square brackets.

We can also search for base forms, e.g. *laki_ OR oikeus_* (17).

(17)
1_§_2 Suomen valtiosääntö on vahvistettu tässä perustuslaissa {perustuslaki_N}.
1_§_3 Valtiosääntö turvaa ihmisarvon loukkaamattomuuden ja yksilön vapauden ja oikeudet {oikeus_N} sekä edistää oikeudenmukaisuutta yhteiskunnassa.
1_§_4 Suomi osallistuu kansainväliseen yhteistyöhön rauhan ja ihmisoikeuksien {ihmisoikeus_N} turvaamiseksi sekä yhteiskunnan kehittämiseksi.
2_§_2 Kansanvaltaan sisältyy yksilön oikeus {oikeus_N} osallistua ja vaikuttaa yhteiskunnan ja elinympäristönsä kehittämiseen.
2_§_3 Julkisen vallan käytön tulee perustua lakiin {laki_N}.

When the left boundary mark { was missing, all base words ending with *laki* or *oikeus* will be printed.

The operator AND is more complicated to implement, because the hits must be on the same line and in the given order. The safest implementation method is to use base form search such as *laki_ AND oikeus_* (18).

(18)
9_§_5 Lailla {laki_N} voidaan kuitenkin säätää, että Suomen kansalainen voidaan rikoksen johdosta tai oikeudenkäyntiä varten taikka lapsen huoltoa tai hoitoa koskevan päätöksen täytäntöönpanemiseksi luovuttaa tai siirtää maahan, jossa hänen ihmisoikeutensa {ihmisoikeus_V} ja oikeusturvansa on taattu.
10_§_4 Lailla {laki_N} voidaan säätää perusoikeuksien {perusoikeus_V} turvaamiseksi tai rikosten selvittämiseksi välttämättömistä kotirauhan piiriin ulottuvista toimenpiteistä.
19_§_2 Lailla {laki_N} taataan jokaiselle oikeus {oikeus_V} perustoimeentulon turvaan työttömyyden, sairauden, työkyvyttömyyden ja vanhuuden aikana sekä lapsen syntymän ja huoltajan menetyksen perusteella.

We can also combine the operators AND and OR. This makes possible quite complex search operations. If our sear key is *laki_ and asia_ or oikeus_*, we get results as in (19).

(19)
9_§_5 Lailla {laki_N} voidaan kuitenkin säätää, että Suomen kansalainen voidaan rikoksen johdosta tai oikeudenkäyntiä varten taikka lapsen huoltoa tai hoitoa koskevan päätöksen täytäntöönpanemiseksi luovuttaa tai siirtää maahan, jossa hänen ihmisoikeutensa {ihmisoikeus_V} ja oikeusturvansa on taattu.
10_§_4 Lailla {laki_N} voidaan säätää perusoikeuksien {perusoikeus_V} turvaamiseksi tai rikosten selvittämiseksi välttämättömistä kotirauhan piiriin ulottuvista toimenpiteistä.
58_§_3 Valtioneuvosto voi tällöin antaa eduskunnalle muusta kuin lain {laki_N} vahvistamista, virkaan nimittämistä tai tehtävään määräämistä koskevasta asiasta {asia_V} selonteon.

The search key can also refer to surface word, such as in *laki_ and oikeus_ or yhteiskunnan* (20).

(20)
10_§_5 (5.10.2018/817) Lailla {laki_N} voidaan säätää välttämättömistä rajoituksista viestin salaisuuteen yksilön tai <yhteiskunnan> turvallisuutta taikka kotirauhaa vaarantavien rikosten tutkinnassa, oikeudenkäynnissä, turvallisuustarkastuksessa ja vapaudenmenetyksen aikana sekä tiedon hankkimiseksi sotilaallisesta toiminnasta taikka sellaisesta muusta toiminnasta, joka vakavasti uhkaa kansallista turvallisuutta. (5.10.2018/817)
19_§_2 Lailla {laki_N} taataan jokaiselle oikeus {oikeus_V} perustoimeentulon turvaan työttömyyden, sairauden, työkyvyttömyyden ja vanhuuden aikana sekä lapsen syntymän ja huoltajan menetyksen perusteella.

4 Counting and listing hits

When found hits are marked, as we see above, we can also produce many kinds of statistics. In Technical Report No. 44 (Hurskainen 2019b) we listed POS statistics of *Perustuslaki* on the basis of base form. The current more advanced implementation makes possible the statistics of full surface word matches with angle brackets, and partial surface word matches with square brackets.

With the key words *lailla OR oikeus_* we get the result as in (21).

(21)
```
 1 aloiteoikeus_N
 9 hallinto-oikeus_N
 2 hovioikeus_N
 5 ihmisoikeus_N
 1 käräjäoikeus_N
91 <lailla>
68 oikeus_N
 6 perusoikeus_N
 1 [perustuslailla]
 1 syyteoikeus_N
 7 valtakunnanoikeus_N
 1 verotusoikeus_N
 1 äänioikeus_N
```

With the key word *lailla* we refer to the surface words. There are 91 full word matches and one partial word match. All other hits are base forms of the key word *oikeus_*. So we found all three kinds of hits and their statistics.

If our both search words are base forms, such as *laki_ OR oikeus_*, the result is as in (22).

(22)
```
 1 aloiteoikeus_N
 9 hallinto-oikeus_N
 2 hovioikeus_N
 5 ihmisoikeus_N
```

```
  4 itsehallintolaki_N
  1 käräjäoikeus_N
  3 kirkkolaki_N
144 laki_N
  1 maakuntalaki_N
  1 maanhankintalaki_N
 68 oikeus_N
  6 perusoikeus_N
 33 perustuslaki_N
  1 syyteoikeus_N
  7 valtakunnanoikeus_N
  1 verotusoikeus_N
  1 äänioikeus_N
```

5 Special problems in implementation

In implementing the search system, there were some difficult problems to solve. One of them was the management of the scandic letters ä, ö, å, Ä, Ö, and Å. There is an incredible number of encoding of these letters, and, when the editor is constructed to guess the encoding in saving the document, the encoding went wrong every now and then. The strangest thing was that the editor frequently chose the Japanese encoding.

To avoid this problem, I chose to rewrite all scands as bigrams in ASCII code and use such letter combinations, which never occur in Finnish. The whole process from the beginning to the end was run with these bigrams, and only in the end the bigrams were returned to scands. This method required that also the search word with scands was first converted to using bigrams.

Another problem was in implementing the use of asterisk in cutting the search string, so that the use would be simple and would not require word boundary marking from the user. The solution was to mark the beginning and end of each surface word with a non-alphabetical character in target text. This made the word boundary visible, and the character could be used in marking the valid hits. It also required that the search word would be converted into the form, that it would match with the target word in disired way. Examples of this process are in (21).

(23)
laki* > &laki
*laki > laki&

When using the asterisk in the beginning or end of the string, the system converts it to the plain character string and places the word boundary character to the opposite side of the string. Now the converted keyword would match in the beginning or end of the target word.

The whole system was constructed so, that the search method using the asterisk was the very first search method. After this the word boundary code was removed from the retrieved text, and all other search methods for modifying the search result were done on this cleaned text. Note that the word boundary marking was only needed for

implementing the search method with asterisk, and all other search methods could be done without explicit word boundary marking.

6 Conclusion

In this report we have shown that it is possible to design precise and covering information retrieval systems, which make use of the analysed representation of text, but which also facilitate the use uf the traditional method based on surface text search. It is also possible to mark the hit types each with separate coding. Such a search system replaces the traditional search systems based on surface string search only.

References

Hurskainen, Arvi, 2019a. Accurate information retrieval using text analysis and disambiguation. *Technical Reports in Language Technology*, Report, No 36. http://www.njas.helsinki.fi/salama/technical-reports.html

Hurskainen, Arvi, 2019b. Two methods for accurate information retrieval. *Technical Reports in Language Technology,* Report, No 44. http://www.njas.helsinki.fi/salama/technical-reports.html