Conquering Language: Using NLP on a Massive Scale to Build High Dimensional Language Models from the Web

Gregory Grefenstette

► To cite this version:

Gregory Grefenstette. Conquering Language: Using NLP on a Massive Scale to Build High Dimensional Language Models from the Web. CICLing, Feb 2007, Mexico, Mexico. pp.35 - 49, 10.1007/978-3-540-70939-8_4 . hal-01081036

HAL Id: hal-01081036

https://hal.inria.fr/hal-01081036

Submitted on 6 Nov 2014

Conquering Language: using NLP on a massive scale to build high dimensional language models from the Web

Gregory Grefenstette

Commissariat à l'Energie Atomique, CEA LIST, SRCI, BP 6,
92265 Fontenay aux Roses Cedex, France
Gregory.Grefenstette@cea.fr
http://www-list.cea.fr/gb/index_gb.htm

Abstract. Dictionaries only contain some of the information we need to know about a language. The growth of the Web, the maturation of linguistic processing tools, and the decline in price of memory storage allow us to envision descriptions of languages that are much larger than before. We can conceive of building a complete language model for a language using all the text that is found on the Web for this language. This article describes our current project to do just that.

1 Introduction

Linguistics has long been a descriptive science. From early prescriptive grammars of Sanskrit, Greek and Latin, through the elaboration of glosses in the middle ages, to the comparison of language families starting in the mid eighteenth century, down to the interest in computational grammars from the mid twentieth century until now. Man has tried to describe the interesting and difficult in language. In the last part of the twentieth century, computers were added into the mix, exploiting large hand tagged text collections, first the Brown and Lancaster corpora, and finally the British National Corpus, with its 100 million words of hand corrected, part-of-speech tagged text. But even this computational linguistics research is ultimately based on the descriptions and choices made by the compilers of the hand tagged corpora. This first round of descriptive linguistics research has led to the creation of large language resources: large lexicons for morphological analysis, training text for part of speech taggers, and robust grammars for analyzing large quantities of text. We are now ready to enter into a second round of linguistics, in which the descriptions of language will no longer be based on manual effort of description, but in which complete descriptions of language use and behavior can be automatically acquired and stored. We can now move from the question *What do we know about language?* to the question *What do we do now that we know everything about a language?*

The convergence of three different phenomena allows us to consider that it is now possible to know everything we want to know about every word in every written language. These three phenomena, which have all appeared in the last decade, are

1. the maturity of the linguistic processing tools mentioned above.
2. the continuing explosion of cheap computing power and storage possibilities
3. the arrival of the internet, bringing free access to enormous quantities of text

Exploiting all three will allow us to see linguistics and language differently.

In this paper, we present our first attempts to create a full description of a language, to conquer the language, in some sense, by extracting and treating as much of the presence of the language on the Web as possible. The paper is divided into the following sections. First we describe how we know how much language is available for a given language, and we show how to gather basic statistics for the words in the language. Next, in Section 3, we detail the process for extracting language modeling information from the internet, estimating the time necessary for generating the entire model. In Section 4, we show some potential applications of the extracted model. This is followed by a description of related research and a conclusion.

2 Estimating language presence

Before we begin deriving a given language model, we must ask whether we have access to enough text to build a complete model for that language. How much text is enough text? We cannot say. But let us take as a minimum the number of 100 million words, the number of words in the largest hand corrected, annotated corpus created for English, the British National Corpus. Suppose that we want to make a language model for Danish, or Basque, or Catalan. Do we at least have 100 million words of these languages available to us through a search engine? We tried to answer this question for a number of languages starting in 1996. Before 2002, search engines such as AltaVista displayed occurrence counts for each query word as well as the global page count for the query.[1] When this real count was available it was possible to estimate the total number of words indexed for language by probing search engines with common words for a language, that each had a known frequency for that language. Details of this estimation are given in [1]. Results from our last estimation of language volumes, using data graciously obtained from Yahoo, gave the following estimates seen below in Table 1. The volume of text available through search engines for all of these languages was growing over the three year period from 2001 to 2004, doubling or more for most languages. In all cases the raw number of words available for these languages

[1] This page count is the only information returned by search engines today. And even this page count is no longer an actual page count, but only an estimation of the number of pages containing the query words. Currently search engines calculate actual page counts on a sample database and extrapolate to the total number of pages they have indexed. See Jean Veronis's blog for a discussion of this estimate. http://aixtal.blogspot.com/2005/02/web-googles-missing-pages-mystery.html

is greater than the number of words available in the British National Corpus[2].The numbers given in this table are also only a lower bound on the number of words available, because other pages of text exist for all of these languages on pages not crawled by the reference search engine, either because of its crawling policy, or because the text is available on the Hidden Web through search form interfaces (such as MedLine's search interface) and not as static pages that can be downloaded and crawled. See [2] for accessing more text in the Hidden Web. We can thus presume that, for many of the languages present on the Web, we have a sufficient quantities of text from which to build languages models.

	Word count estimate 2001	*Word count estimate 2004*
Basque	55,340,000	148,776,424
Estonian	98,066,000	208,739,164
Croation	136,073,000	188,527,817
Catalan	203,592,000	1,206,027,725
Finnish	326,379,000	826,416,488
Danish	346,945,000	1,684,667,584
Czech	520,181,000	1,008,251,069
Dutch	1,063,012,000	3,333,039,454
French	3,836,874,000	13,648,627,968
German	7,035,850,000	16,583,288,838
English	76,598,718,000	145,959,354,990

Fig. 1. Language estimates for a few European languages in 2001 and 2004. The number of words available on the Internet doubled or more for most of these languages in this period. All of the languages have more than 100 million words, the number of words in the British National Corpus

Now let us suppose that we begin with a full list of words for the language, excluding proper nouns and technical terms. Such lists can be obtained by generating all word forms using a morphological analyzer, or from other sources, such as the Ispell data files for the language [3].

In the rest of this paper, we will use French as a sample language, though the results should be the same for any other non agglutinative language.[3] Beginning with a list of French word forms generated by the LIMA system [4], we have a list of 400,000 French word forms to model, generated from just under 100,000 lemmas. For example the lemma *chien (dog)* produces two words in the word form list *chien* and

[2] Of course, the British National Corpus is marked up in a variety of ways: part of speech tags, dates published, speakers, etc. Internet text has less and more disparate metatags, if any.

[3] Even agglutinative languages such as Finnish and Hungarian can be used, if one accepts a restriction on the number of letters any word in the word list generated, or used. Though, in this case, the number of word forms will be in the millions rather than the hundreds of thousands.

chiens. The verb *aimer (to love)* produces all the conjugated forms *aime, aimes, aimons, aimez*, etc. Though verbs generate many word forms, nouns usually only generate two in French so the average number of word forms per lemma here is about 4.

We take this word list, and as a first step in building a language model for French, we gather counts of the number of web pages that each word appears in. Since it is relatively common for a given word in languages with common historical roots to share word forms (for example, both English and French use the word form *relations* among thousands of other cognates), we need some way to restrict searching ambiguous words to obtain counts specific to our target language. Though most search engines allow the searcher to restrict their search to pages in one of a number of specified languages, we decided not to use this option for two reasons. One, we do not know what language identification algorithm [5][6] the search engines use for deciding whether a page is written in a given language, these algorithms being unpublished and unevaluated. Second, we wanted to develop a technique for all web languages, not restricted to those on a search engine list.[4] In order to restrict the count of pages to pages written in French, we decided to add on a number of common French words as "anchors" for any query, the idea being that a page that contains our search word AND these anchor words is probably in French. We chose the following anchor words in French: {*et, le, la, que, pour*}. So when we want to query the web for the number of pages that contain the word *relations*, for example, we build the following search query:

"relations" "et" "le" "la" "que" "pour"

For English, we would a query like:

"relations" "the" "with" "and" "in" "of"

Of course, in either case, it is possible to retrieve pages with the above queries that are neither in French, nor in English. And also, we know that we will be missing some French pages that contain the word *relations* but which do not contain one of the five anchor words. To test the presence of these "anchor" words in a known language corpus, we took a list of documents from the TREC information retrieval campaigns. In the French TREC corpus, there are 62,464 documents with more than 200 words. 12,031 of these documents (19%) do not contain all five of the French anchor words we used. For English, using a set of 13,856 documents with more than 200 words from the corpora in TREC, we found 1359 documents (10%) that did not contain the English anchor words given above. As the length of the document increases, the percentage of documents that do not contain the anchor terms decreases. Since we are looking to construct a language model for the whole language, we are more interested in longer documents, using this anchor technique provides counts that are probably inferior, but in a systematic way, to the real counts of words, and tend to return longer documents.

If we run the query *"relations" "et" "le" "la" "que" "pour"* on Google on a given day, we find 3,880,000 pages that contain all six words. If we run the query *"relations"* without the anchor words on Google but with the "Search French pages"

[4] For example, Google allows the user to search in 36 languages. The restriction of search results to a given language should not be confused with the option to use different languages in the search interface. Google provides more than 115 interface languages.

option activated, then we get 4,600,000 result pages. If we run the same queries on AllTheWeb, the result page says that there are 12 million pages with the anchor words and 18 million for *"relations"* (without the anchors) in the AlltheWeb French documents[5]. We see that these numbers are a rough approximation. Jean Veronis explains that these numbers must be taken with precaution (see footnote 1 above). Here, though, we through caution to the wind, since we are not interested in exact number but only relative magnitudes, and we continue to collect page counts for each of the 400,000 French words using this anchoring technique. This harvesting can be done over a period of 400 days using, for example, the Google search API which allows a user to send 1000 queries a day. If this API is spread over more than one machine and more than one user, then the searches may be parallelized. Other search engines provide other means for sending batched queries. After a certain number of days, the first element of the language model can be acquired: the relative frequencies of all the words for a language on the web (under the unknown indexing bias of the search engine used). This step has been performed for a large subset of French words in the Lexique project [7], and can be downloaded from www.lexique.org.

héliastes	**445**	**hélicoïdes**	**275**
hélice	**211000**	**hélicon**	**1910**
hélices	**121000**	**héliconienne**	**9**
hélichryses	**10**	**héliconiens**	**10**
héliciculture	**842**	**hélicons**	**108**
hélicicultures	**7**	**hélicoptère**	**723000**
hélicier	**48**	**hélicoptères**	**535000**
héliciers	**14**	**hélicos**	**65200**
hélico	**143000**	**héligare**	**73**
hélicoïdal	**13800**	**héligares**	**8**
hélicoïdale	**13700**	**hélio**	**17300**
hélicoïdales	**1150**	**héliocentrique**	**3380**
hélicoïdaux	**8680**	**héliocentriques**	**482**
hélicoïde	**421**	**héliocentrisme**	**1980**

Fig. 2. A subset of the French lexicon with sample page counts from a search engine. These are the page counts we acquired for pages containing the given word and the following anchor words "le" "la" "et" "que" "pour", used to *anchor* the page in French.

[5] On both search engines, if we search for *relations la la que et pour* with the "English Only" language option activated we still get hundreds of thousands of documents back, though the snippets look obviously French. This is one more reason why we do not want to trust the unknown language identification methods of search engines.

A sample of the type of word counts, using anchor words, can be seen in Figure 2. If we retain up to one hundred URLs for each of the 400,000 French word forms, we gather about 5 million distinct URLs as a basis for creating our language model.

3 Extracting Language Modeling Information

Once we are convinced that we have enough text for the language, and we have a full list of words (surface forms) for the language, we can begin to build the language model of the language as shown in the previous section. While collecting information about the frequency of each word, we can also collect a seed list of URLs for each word. We are not interested here in indexing the web, that is, associating URLs to each word in the lexicon, but rather collecting examples of word usage. Nonetheless, we do have to perform a crawl of a large portion of the web. Each page retrieved then has to be treated by language specific natural language processing tools (for example [8]) to extract the model. This section describes all these steps.

3.1 Fetching web pages and Character encoding

When we collect text for non English languages, text encoding becomes a problem to face. Both HTML and XHTML markups provide tags for specifying the character set used in a web page:

```
<meta http-equiv="Content-Type" content="text/html; char-
                          set=UTF-8">
         <?xml version="1.0" encoding="ISO-8859-1"?>
```

But in practice we find that we could not always trust human encoding of this attribute. We sometimes have to test and discover the actual character encoding ourselves. This encoding detection can be done in conjunction with language identification [9] or separately. We performed it separately. For encoding, we implemented a simple system that takes as input a URL, fetches the HTML of this URL using the UNIX command *wget*, and then extracts the encoding found in the HTML metatags. If this tag is found, we use the UNIX command *recode* to transform the character encoding specified into UTF8, which is the input code expected by our natural language processing tools. However, we noticed in dealing with French web text, that pages specified as using the iso-8859-1 character set (formerly the most common encoding for Western European languages) were often actually encoded in UTF-8 (the new standard). We modified our fetching program so that when HTML pages specify this encoding, we re-check using the UNIX command *file* which identifies the type and encoding of a file. If *file* finds that the file is in UTF, we replace the encoding specified in the HTML text. If no charset is specified, we suppose that it is written in iso-8859-1 which is the most common encoding for French. Obviously, if we were to treat another language instead of French, this strategy would have to be adapted. Just as knowledge about the best anchor words to use is needed for a new language, we also have to know the most common encoding for each language[6].

3.2 Extracting Text from Web Pages

Once we have recoded the fetched web page in UTF-8, we use another command *lynx*[7] that formats a web page for a line-based (*tty* or teletype) device. This command was useful in the first days of the Web in the early 1990's when line-based computer terminals were still present, and it is still useful for browsing the web in *xterm* windows, and maintained. For web text processing, the *–dump* argument to *lynx* is particularly useful since it extracts only the visible text from a web page, eliminating all the HTML markup and dealing with issues such as frames. Since *lynx* relies heavily on specified character set encodings, we pass it as input our wget-fetched and UTF-8 re-encoded HTML that we have prepared, as described in the last section, rather than the raw URL. *lynx* also provides a formatted listing of all the outgoing URL links found in a web page.

3.3 Language Identification and Web Crawling

Beginning with the seed URLs for each word in the language, we first verify that we have not already treated this URL by comparing it to a list of already seen URLs. If the URL is new, we apply the text extraction steps specified in the last two sections[8]. It is still possible that the input file is not really in the language that we are interested in modelling. The URL may have been found because it contained a target-language word and all of the anchor words, but still be in a different source language. For this reason, we submit the extracted text to a language identifier. Language identifiers use characteristic sequence of letters or short words [6] to guess which language a text is written in. We use the language identifier of the LIMA natural language processing system, and reject the page if the most likely language is not the one we are interested in, French in this paper.

When a text has been identified as being in French, we also extract all the outgoing links from the page and add them to a list of new pages to browse. The two lists of seen URLs and new URLs are the basis of any web crawler [11].

Having implemented this simple crawler for extracting French text using a Full French lexicon, anchor words, and language identification, we find that about 75% of the URLs that we access produce useful text. The remaining 25% percent of URLs fail for one of the following reasons:

[6] See http://www.cs.tut.fi/~jkorpela/chars.html for a tutorial on character set encodings

[7] http://lynx.isc.org/ is the lynx homepage

[8] We do not consider the problem of different pages with the exactly same content. Studies performed by [10] found an incidence of less than 5% of exact page matches with different URLs during a collect of web pages.

- page no longer exists, or timeout in accessing the page. We give each URL 3 seconds to respond.
- no text on page. Page only contains images, or pointers to other pages.
- text not in French, according to our language identifier

Each URL found that each French URL successful retrieved reveals on average 6 new URLs to explore. The average size of a French retrieved web page converted into text is 8000 bytes.

3.4 Natural Language Processing

For each French text gathered, we apply the LIMA natural language processor [4] to extract the elements of the language model we want to create for French. Our linguistic analysis[9] performs tokenization, morphological analysis, part-of-speech tagging, and dependency extraction, creating an analysis graph that can be traversed to extract many different types of output. We extract the following elements for each word:

- the lemmatized form of the word
- the syntactic dependency relations involving the word
- the normalized noun phrases, in format LIMA (lemmas, stopwords[10] removed)
- the words and phrases found in a window of 5, and 10 non stopword words
- named entities found in a window of 20 words around each word.

For example from the text.

```
Vérifiez encore l'angle du robinet et serrez les boulons
(verify once again the spigot angle and tighten the nuts)
```

we extract the lemmatized words:

```
vérifier, angle, robinet, serrer, boulon
```

the dependency relations :

```
COMPDUNOM(robinet,angle)  noun-modifier(spigot,angle)
COD_V(angle,vérifier)     verb_object(verify angle)
COD_V(boulon, serrer)     verb_object(tighten nut)
```

the nomalized noun phrase:

[9] Linguisitic analysis tools can be found at http://www.alphaworks.ibm.com/tech/lrw/ and http://www.gate.ac.uk/download/

[10] A stopword is a function word such as an article, preposition, etc; that is not usually indexed in information retrieval systems. http://www.ranks.nl/tools/stopwords.html provides pointers to lists of stopwords Catalan, Czech, Danish, Dutch, English, French, German, Hungarian, Italian, Norwegian, Polish, Portuges , Spanish, and Turkish.

angle_robinet *(spigot angle)*

the words and phrases found in a window of five words around each word. Each lemma in this two column list is listed with another lemma that is found 2 (non-stop) words or concepts, before or after the word :

```
vérifier angle                  robinet serrer
vérifier angle_robinet          robinet boulon
angle vérifier                  serrer angle_robinet
angle angle_robinet             serrer robinet
angle robinet                   serrer boulon
robinet angle                   boulon robinet
robinet angle_robinet           boulon serrer
```

Similarly, we extract the lemmas and phrases in a window of 5 words before and after each word, as well as the named entities (predefined categories of phrases such as the names of persons, organizations, and locations, see [12]) that are found up to 10 words before or each word.

3.5 Processing Output and Example

This output generates a lot of data for each page treated. For an average size page of 8 Kb of text, 300 Kb of output data is produced. We are currently examining efficient ways of storing this data. We are currently in the process of collecting the data from the Web for the 5 million seed URLs for French. Time estimations on the subset we have treated are that we can produce output for 400 URL per hour on one PC (Intel Xeon, CPU 3.40GHz, 2Mb cache), with fetching and converting the text for a URL only accounting for a small part of the time, and the linguistic processing and extraction from the results graph accounting for the rest. This means that we can treat 1 million URL in 100 days. Data extraction is independent for each URL, so that process can be parallelized. And we are currently planning to move to a parallel machine.[11]

The final model of a word will depend on the amount of text treated. As more of the web is covered, the model will reach a web-biased stasis. And we hope that the results approach some satisfying model of language for the applications we will sketch below.

In the meantime, we can show some preliminary, anecdotal results. After treating a sample of around 30,000 URLS, we find that the following type of information for a common word like *avion (airplane)*

Common verbal patterns for *avion (airplane,plane)* with their frequencies:

[11] We are planning to move to the TERA-10 supercomputer. This CEA LIST machine, made by Bull, is composed of 4352 Intel Montecito dual-core processors (8704 cores), connected together by a Quadrics high-performance interconnection network and can perform 50 teraflops (50,000 billion operations per second).

COD_V(avion , prendre) (737)	*take an airplane*
SUJ_V(avion , décoller) (115)	*airplane takes off*
SUJ_V(avion , atterrir) (82)	*airplane lands*
COD_V(avion , fabriquer) (80)	*build an airplane*
COD_V(avion , voir) (75)	*see a plane*
SUJ_V(avion , survoler) (74)	*plane flies over*
SUJ_V(avion , aller) (73)	*plane goes*
SUJ_V(avion , arriver) (68)	*plane arrives*
SUJ_V(avion , voler) (61)	*plane flies*
COD_V(avion , reprendre) (59)	*take plane again*
COD_V(avion , détourner) (59)	*hijack plane*
COD_V(avion , piloter) (56)	*fly a plane*
SUJ_V(avion , venir) (54)	*plane comes*
COD_V(avion , utiliser) (54)	*use a plane*
COD_V(avion , abattre) (54)	*shoot down a plane*

Common phrases involving *avion(airplane,plane)* as a modifier

billet_avion (671)	*airplane ticket*
accident_avion (95)	*airplane accident*
pilote_avion (76)	*airplane pilot*
détournement_avion (67)	*airplane hijacking*
descente_avion (66)	*airplane landing*
voyage_avion (58)	*airplane trip*
vol_avion (57)	*airplane flight*
place_avion (47)	*airplane seat*
passager_avion (46)	*airplane passnger*
bruit_avion (45)	*airplane noise*
crash_avion (40)	*airplane crash*
retard_avion (39)	*plane delay*

common phrases involving *avion(airplane,plane)* as a head of the phrase:

avion_ligne (178)	*commercial plane*
avion_combat (92)	combat plane
avion_militaire (83)	*military plane*
avion_petit (81)	*small plane*
avion_chasse (80)	*fighter plane*
avion_premier (75)	*first plane*
avion_transport (51)	*transport airplane*
avion_réaction (43)	*jet airplane*
avion_civil (43)	*civilian airplane*
avion_américain (43)	*American plane*
avion_privé (38)	*private plane*
avion_hélice (34)	*propeller plane*

If we consider the words found in a window of 5 words before or *after avion (airplane)* in these web pages, we get a list that begins with prendre (2602), billet (2580), pouvoir (2392), faire (2186), aller (1864), aéroport (1820), vol (1510), devoir (1324), voir (1118), arriver (970), pilote (950), passager (894), ...

But since we know the relative frequency of each word for French as seen in section 2. It is more interesting to look at the words that are most strongly associated with *avio (plane)*, rather than seeing those that are most common, as given above. For this we can now calculate the mutual information of each word and plane, since we know how many times the word appears on the Web inFrench, and how many times it is found with our target word (here *avion*) in the Web pages we have treated, as suggested by [13]. The first two hundred most strongly associated words with *avion(plane)* from these URLS are

abattre accident acheter aile aimer air aller altitude amener américain annoncer apparaître appareil appeler aérien arme armée aéroport arrivé arrivée arriver arrêter attendre atterrir atterrissage avancer aviation bagage bateau billet bombe bord bruit bus cabine cargo chambrer char chasse ciel civil combat commencer commercial compagnie complet construire continuer contrôleur crash croire décider déclarer décollage décoller demander descendre descente destination devoir diriger distance dormir départ déplacement détourné détournement détourner détruire effectuer embarquement embarquer emmener empêcher ennemi entendre environ escale essayer exister expliquer exploser explosion fabriquer faillir faire finir flotte frapper gêner heure hélice hélicoptère horaire hôtel indestructible indiquer israélien laisser lâcher léger maintenance manquer marcher matin matériau militaire missile monder monter moteur navire noir nuit observer occuper parler passager passeport passer payer penser perdre permettre petit pilotage pilote piloter piste plaire porter pouvoir prendre provenance prévoir équipage quitter réaction radar ramener rappeler rater reconnaissance regarder rejoindre remplir rentrer repartir reprendre rester retard revenir rouler route réserver russe sauter sembler sentir serpent siège séjour sol sommer sortir suivre survoler taire taxi étayer température tenter terroriste tomber toucher tour tourner trafic train trajectoire transport transporter utiliser venir véhicule vitesse voisin voiture vol volant voler vouloir voyage voyager

This set of words will change as more text is extracted for this word (*avion*) though we can imagine that many of the words will still be very strongly associated with the concept.

Now imagine that we have these common phrase, and syntactic patterns, and ‘clouds of words’ (that also have frequencies associated with them that are not shown above) for each of the 100,000 lemmas of French. We plan to have this resource at the end of the first year of this project in 2007. We will sketch some of the uses of this vast language model in the next section.

4 Possible Applications

The uses of a language model are varied. The model that we are building will provide the relative frequency of all language phenomena for the common set of 100,000 French lemmas, up to the dimension of the Web that we treat. Relative frequency is one key of word importance [14] that has long been used in information retrieval (in the guise of inverse document frequency). And information on lexical associations has been used to resolve structural ambiguities in parsing [15][16], but never based on complete explicit language statistics from the web, though the implicit web statistics have been used [17].

For other language applications, such as machine translation and speech understanding, the benefits of having a Web-sized model of language are even more evident. Implicit web statistics have been shown to be useful in choosing the right translations of noun phrases [18], translating and transliterating proper names [19][20].

Current text-to-speech system make errors such as the following which is output by one of the leading systems in the world:

Text spoken:

le pape est apparu très fatigué il a célébré l' eucharistie le dernier repas du christ mais il a renoncé au lavement des pieds *(the pope appeared very tired. he celebrated the eucharist, the last meal of christ, but he renounced performing the washing of the feet.)*

Output of Speech-to-Text:

le pape est apparu très fatigué il a célébré le péristyle le dernier repas du christ et mais il a renoncé au lavement didier *(the pope appeared very tired. he celebrated the* ***peristyle****, the last meal of christ, but he renounced performing the washing of* ***didier****.)*

Both interpretations are available inside the speech-to-text system, but the current language model that it uses is based upon ngrams sequences of words and analysis can break down when unseen sequences appear. In our explicit model of all the words in the language, we will have statistics about the common dependency relations such as

SUJ_V (eucharistie, célébrer) *celebrate eucharist*
NNPREP (pied, lavement) *noun-modifier(feet,washing)*

These elements of the language model will allow us to prefer more common interpretations of speech streams and produce more likely translations of speech into text.

There are many other language applications that use Web data to extract knowledge, for detecting affect and opinion [21], gathering world knowledge about visual aspects [22], and gathering more general knowledge [23]. But all these techniques current employ techniques for probing the Web for word statistics rather than building an explicit linguistically treated model, as suggested in [24].

5 Related Research and Conclusion

Google announced in the summer of 2006 that it has just extracted all sequences of 5 words appearing more than 40 times in over 1 trillion words of indexed text[12]. It is being distributed through the Linguistic Data Consortium on 6 DVDs. There are 1.1 billion sequences with their frequencies listed. Their research has shown that the more data you apply to tasks such as machine translation, the larger the language models, the better the results[13].

It is obvious that online dictionaries are not the answer to the problem of creating the next generation of language models. Dictionaries contain descriptions of the extraordinary, what a person might need to know about a word or phrase, whereas the computer, for its language understanding needs to know the ordinary: what words are found in what patterns, what other words are to be found nearby. We are now at a point where it is possible to extract all this information on a very large scale from the sum of what humanity is publishing on the web, creating a very language model as could only have been dreamed of a few years ago. Once this modelling is done for individual words, the next step is to do it for all the lexical structures, with one entry for each multiword noun phrase (for example, performing the same analysis as is shown above with *avion(plan*e) for *billet-avion (plane ticket))* and for each syntactic dependency pair. This is another level of complexity but it is also feasible with current linguistic processing techniques and current computing power.

5 Acknowledgements

This research is being partially funded by a grant from the Fondation Jean-Luc Lagardère http://www.fondation-jeanluclagardere.com and by the NoE MUSCLE (Multimedia Understanding through Semantics, Computation and Learning) Contract EU FP6-507752

[12] This list of 5grams is available on 6 DVDs through the LDC.. http://www.ldc.upenn.edu/Catalog/nonmem_agree/Web_1T_5gram_V1_User_Agreement.html. Google is providing it free for research institutions and universities.

[13] http://webspace.isi.edu/mt-archive/MTS-2005-Och.pdf

References

1. Grefenstette, G., Nioche, J.. Estimation of English and non-English language use on the WWW. In Proceedings of RIAO (2000)
2. Ipeirotis, P. G., Agichtein, E., Jain, P., and Gravano, L.: To search or to crawl?: towards a query optimizer for text-centric tasks. In Proceedings of the 2006 ACM SIGMOD international Conference on Management of Data (Chicago, IL, USA, June 27 - 29, 2006). SIGMOD '06. ACM Press, New York, NY, (2006) 265-276
3. Nemeth, L., Tron, V., Halacsy, P., Kornai, A., Rung, A., Szakadat, I. Leveraging the open source ispell codebase for minority language analysis, In Fir:st Steps in Language Documentation for Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation, Proceedings of the SALTMIL Workshop at LREC, (2004) 56–59.
4. Besançon, R., de Chalendar, G., Ferret, O., Fluhr, C., Mesnard, O., Naets, H.:. Concept-Based Searching and Merging for Multilingual Information Retrieval: First Experiments at CLEF 2003. Proceedings of CLEF (2003) 174-184.
5. Cavnar, W. B., Trenkle, J. M.: N-gram based text categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, (1994) 161-175
6. Grefenstette, G.: Comparing two language identification schemes. In Proceedings of the Third International Conference on the Statistical Analysis of Textual Data (JADT'95), Rome, December 11-13 (1995) 263-268
7. New, B., Pallier, C., Brysbaert M., Fer L.: Lexique 2 : A New French Lexical Database. Behavior Research Methods, Instruments, & Computers, Volume 36, Number 3, (2004) 516-524
8. Cunningham, H.., "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications," *Proc. 40th Anniversary Meeting Assoc. for Computational Linguistics* (ACL 2002), Assoc. for Computational Linguistics, East Stroudsburg, Pa., (2002)
9. G-I. Kikui. Identifying the coding system and language of on-line documents on the internet. In Proceedings of the 16th International Conference on Computational Linguistics (COLING) (1996)
10. Berland, S., Grabar, N. : Assistance automatique pour l'homogénéisation d'un corpus Web de spécialité. In Actes des 6èmes Journées internationales d'analyse statistique des données textuelles, JADT 2002, Saint-Malo (2002)
11. Heydon, A., Najork, M.: Mercator: A scalable, extensible Web crawler, World Wide Web, v.2 n.4, (1999) 219-229
12. Sundheim, B.: Overview of results of the MUC-6 evaluation. In Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference held in Columbia, Maryland, November 6-8, (1995) 13-32
13. Church, K. W. and Hanks, P.: Word association norms, mutual information, and lexicography. *Comput. Linguist.* 16, 1 (Mar. 1990), 22-29
14. Hiemstra. D.: A probabilistic justification for using tf.idf term weighting in information retrieval. International Journal on Digital Libraries, 3(2): (2000) 131-139
15. Hindle, D. and Rooth, M:. Structural ambiguity and lexical relations. In Computational Linguistics, 19(1), the MIT Press (1993) 103-120
16. Merlo P., Crocker M. W., Berthouzoz C.: Attaching multiple prepositional phrases: Generalized Backed-off Estimation. In Cardie C. and Weischedel, R. editors, Proceedings of the second conference on Empirical Methods in Natural Language Processing, EMNLP-97, (1997). 149-155

17. Nakov, P., Hearst, M.: Using the Web as an implicit training set: Application to structural ambiguity resolution. In: Proceedings of HLT-EMNLP, Vancouver, British Columbia, Canada (2005) 835–842
18. Grefenstette, G.: The World Wide Web as a resource for example-based machine translation tasks. In Proceedings of the ASLIB Conference on Translating and the Computer, London (1998)
19. Yiping Li and G. Grefenstette. Translating Chinese idiographic characters via corpus and web validation. In CORIA'2005, Grenoble, France, March 9-11, 2005
20. Qu, Y., Grefenstette, G.: Finding Ideographic Representations of Japanese Names Written in Latin Script via Language Identification and Corpus Validation In Proc. of ACL (2004) 184-191.
21. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association, ACM Transactions on Information Systems (TOIS), 21 (4), (2003) 315−346
22. Grefenstette, G.:The Color of Things: Towards the automatic acquisition of information for a descriptive dictionary in Revue Française de Linguisitque Appliquée, vol X-2 1386-1204, (2005) 83-94
23. Cimiano, P., Staab, S.: Learning by googling, ACM SIGKDD Explorations Newsletter, v.6 n.2, p.24-33, December (2004)
24. Kilgarriff, A.: Linguistic search engine. In Kiril Simov, editor, Shallow Processing of Large Corpora: Workshop Held in Association with Corpus Linguistics 2003, Lancaster, England. (2003)