Human Language Technology: 
The Babel Fish

Björn Gambäck  
gamback@sics.se

November 1999

Information and Language Engineering Group  
Swedish Institute of Computer Science  
Box 1263, SE–164 29 Kista, Sweden

and

Computational Linguistics  
Department of General Linguistics  
P.O. Box 4, FIN–00014 University of Helsinki, Finland

Abstract:  
The essay describes some of the main problems which meet us when trying to process human language on a computer. The overall approach is to look at what we would need to do in order to be able to build a device with the same general functionality as Douglas Adams’ Babel fish. That is, a device which can take utterances spoken in one language and instantly translate them into speech in some other language.

Also Published: A shorter version of the essay (in Finnish) appears in Nukkutilmien laskennalliseen mallintamiseen (Viewpoints on Computational Modeling). Center for Scientific Computing, Helsinki, Finland.
The Babel Fish

"'The Babel fish,' said The Hitch Hiker's Guide to the Galaxy quietly, 'is small, yellow and leech-like, and probably the oddest thing in the Universe. It feeds on brainwave energy received not from its own carrier but from those around it. It absorbs all unconscious and mental frequencies from this brainwave energy to nourish itself with. It then excretes into the mind of the carrier a telepathic matrix formed by combining the conscious thought frequencies with nerve signals picked up from the speech centres of the brain which has supplied them. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language. The speech patterns you actually hear decode the brainwave matrix which has been fed into your mind by the Babel fish.'"


Stick a small fish in your ear and understand all languages? Sounds pretty useful! Let's try and go ahead and build such a Babel fish. What do we need? Well, first some fins, yellow fish-skin and such. Hmmm... Actually I don’t think I want to have a live, slimy fish in my ear. Let alone a creature which feeds on brainwaves!

Let’s just settle for trying to build a device with the same general functionality as the Babel fish. That is, a device which can take utterances spoken in one language and instantly translate them into speech in some other language. What do we need to be able to do that, then?\(^1\)

---

1The rest of this report will thus describe some of the main problems which meet us when trying to process human language on a computer. As the alert reader already has noticed by now (I hope :-| the language I'm writing in is English (for a discussion on trying to identify languages, please read on in the main text)). Unfortunately, a text like this one will more or less by force have to contain quite a few terms which you might not have seen before. Some of these words are explained in the text, others I've collected in a short word-list which I've included at the end. In the text itself, those words are marked with an asterisk (*).

Tässä huvossa keskiyön pienen joukkoon sellaisia hankalia ongelmia, joita tukee eteen, kun yritetään käsitellä ihmiskieltyä tietokoneiden avulla. Kuten tarkkaavainen lukijani on jo (toivottavasti :-) huomannutkin, kirjoitan tätä telstää englanniksi (kielen tunnistuksesta lisää itse telstissä!). Käytän englantia, koska suomen kielen taitoni on jokseenkin olematon. Valitettavasti täällä kielissä on aina mukana myös jokin sellainen sana, jota et ole eläköä aiemminkin kuullut. Osa näistä selitetään itse telstissä; osa löytyy tekstiin loppumaa liitetyiltä sanalistalta. Sanalistalta löytyvät sanat olen merkinnyt tähdelä (*).
Firstly, a part which recognizes the spoken utterances and can distinguish which words actually are present among the sound waves which your ears pick up. Then something which extracts the meaning of the sentences — something which tries to identify what the speaker’s intentions* were with making the utterance. Having gone that far, we need to translate this intended meaning into a the way it would be phrased in another language. And finally we need to speak it out again, of course.

Speech Recognition*

Ok, that’s a bit too vague, but at least the general course of action seems clear. So, let’s be a bit more precise. Recognizing speech involves both the function of the regular microphone, as well as a lot of pattern matching and search. This is since the most common way of building a speech recognizer is to record vast* quantities of speech and then try to match the pattern of the speech signal which the microphone registers with some of the old speech we already have in the recorded database. Certainly, if the database consists of full sentences (or maybe even full dialogues*!), the likelihood that we’ll find a match is pretty bad. So we can be a bit smarter and chop* up the speech we have in the database into smaller fragments, for example words.

However, the chances that we’ll find a perfect match are still not that good. Especially, if the database we have consists of utterances from one set of speakers and the person we’re listening to isn’t even in that set! The strength and frequency of speech vary quite a lot from man to man or woman to woman — and even more between men and women... So we need to have some way of describing that difference, if trying to match an incoming signal with an entry in the database. In addition, the speed of the speech varies, so we need a function for manipulating* (compress/extend) the speech along the time dimension, too.

While doing all this manipulating of the speech signal, we need to search our database with old, recorded speech for matches: The search problems involved in speech recognition are tremendous — and the task isn’t exactly made any easier by the fact that it has to be done fast. Yes, as fast as the words are spoken, really (so in real-time).

To help up matters a bit, we can collect data on how the spoken language normally behaves, on which words tend to follow one another, for example, or in other words, to look at the context*. It is the usage of the context information that makes it possible for humans to determine the grammatical categories of the “nonsense” words in the poem Jabberwocky (from the book
Through the Looking-Glass, and What Alice Found There by Lewis Carroll). Most readers would for example induce\(^*\) that toves is a noun and gyre a verb:

“Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe.”

Building our Babel fish, we would also have another problem, language identification. If we collect vast amount of data on the frequencies at which different types of words could follow each other (cooccur) together in one language, there is nothing that says that these relations will hold for another language. For example, the English article ‘the’ can hardly be followed by the name of a person (‘the Julie’?!). This is also true for Swedish (‘den Julian’?!). However, in German this type of construction is very common (‘die Julia’) and in Greek it is almost compulsory (‘i Iolia’).

So, we need to collect word cooccurrence frequencies for all languages our fishy device should be able to translate. But if some weird-looking creature from outer space steps up to us and starts talking, we’ll still not know which of these language models to use in our speech recognizer. Nor would we know which set of words, which vocabulary\(^*\) we should try to recognize. Solving this problem, we could again look at statistical data. If you, for example, would see a piece of text containing a lot of ‘z’ and another one containing a massive amount of ‘x’, you would probably be on a good way to guessing that the first text was written in Polish and the second one in Basque. Most languages have such properties which (more or less obviously and easily) distinguish them from other languages — and not just when it comes to using some specific characters frequently in the written language, but of course also when it comes to using different sounds more often in the spoken language.

**Speech Synthesis\(^*\)**

At the other end, producing (or synthesizing) speech is what is done when we already have decided on which words to say and want to speak them out in a way which sounds natural. There is more to this than meets the eye (or ear, rather!). When we speak, we do not say isolated words, but streams of connected words. In fact, natural speech contains pretty few pauses, at least if you compare\(^*\) it with the spaces found between the words in written
language. Deciding on when to make a pause when speaking is difficult. And the effects of not making a pause are also tricky: When there is no pause between two words, they influence one another. This is called coarticulation and occurs between all two pairs of sounds (phonemes) occurring together, that is, both in-between words and inside the words themselves. In order to get at this, a speech synthesis system would have to be equipped with a large number of rules, or (more commonly) with a database containing all sounds as they appear in the context of other sounds.

Another way to tackle speech synthesis is the strategy often used in telephone services (e.g., “Neiti Aika”), to record even larger chunks of speech (not just phoneme pairs, but entire phrases) and then “glue” together these large units. This is a rather unflexible approach since we must know beforehand the main parts of what the system should say. So this strategy is not a good option for our Babel fish. However, there are some advantages to using such ‘canned’ speech, too: it solves one of the main problems in speech synthesis, namely to give the utterances the correct intonation (prosody). It is hard to know where on the words to put the stress — and which words to stress more than others.

Language

Now, the area of human language technology can in general be said to consist of two major subfields, the Speech oriented topics just discussed, and what’s normally just referred to as Language, which basically means everything else having to do with human languages (and the way these may be processed by a computer). Language itself is certainly the most important means of human–human communication. But even though the human language has been claimed to be the corner-stone of our civilization, the main difference between man and beast, etc., it has received an unproportionally small amount of attention as a channel for human–computer communication. The failure of some language processing projects in the early days of computer science has certainly played a big part in this process, together with the computer science community’s deep-rooted aversion to programming languages which have been claimed to be “natural-english-like” (such as ‘Cobol’ and ‘Basic’).

Still, the goal of computational processing of human language has been a driving force for research from two different directions, by linguists in a subfield called ‘Computational Linguistics’ and by computer scientists in the area of ‘Natural Language Processing’ — the name in itself indicating the contrast to the languages more commonly processed by computers, the
artificial ones constructed specifically for use by the machines. In fact, the term ‘natural language’ is strange in many respects. Primarily, since it refers to what most people would simply call ‘language’, but also since it indicates that the human languages should be “naturally occurring” in some sense. This is certainly not the general case. So are, for example, ‘Esperanto’ and ‘Interlingua’ artificial human languages (while ‘Klingon’ and ‘Quenya’ are artificial languages supposedly spoken by extra-terrestrial beings).

The term confusion left aside, it should be obvious that if a system doing something like “fully-automatic high-quality machine translation” or “general language interfacing to databases” could be built, it would have vast potentials*. Not only economical, but also social in that it would allow non-experts to take advantage of the “information technology revolution”. People who are now often left out from it by the barriers raised by the experts in the field.

Linguistics*

“Aha!” you say. “So language processing is basically the same as Computational Linguistics? Ok, for the word ‘Computational’. It has to do with computing things (and doing that on a computer). But what on Earth is ‘Linguistics’, then?” Checking it out in a dictionary doesn’t give us much help, either:

Linguistics, n. [Cf. F. linguistique.]
The science of languages, or of the origin, signification, and application of words; glossology.

Webster’s Revised Unabridged Dictionary

But at least it tells us that we’re talking about the study of languages here. And studying a human language (say French or Mordvin or Sindangan Sub-anum) is certainly different from studying Language as a concept in itself. That is, trying to find out things which generally hold between a range of languages — or even between all the (more than 5,000) languages spoken on our planet (not to mention languages possibly spoken on other planets...). The study of such linguistic universals has been the main topic of most research in Linguistics during this century.
Building the Babel Fish

So, let’s go back to our initial task, that of trying to build a device which could mimic the functionality of Douglas Adam’s Babel fish. Assuming the speech recognition and speech synthesis are in place, the rest of the stuff we need would fall inside the domain of Computational Linguistics. The first thing we would want to do is to take the words which the speech recognizer has produced and try to deduce some meaning from them. Doing this is a multistage process which would involve the steps shown in the middle of the following picture:

We’ll go through the steps in the picture in turn. After having identified the input words, we would still like to go to work at the parts of the words. The field of morphology looks at the ways the words are built up. A word as we see it consists of a root and a number of affixes. The root is the form you would normally find in a dictionary, the part which carry the most of the particular meaning of the word. The affixes are small inflection parts which come in several different flavours, depending on whether they attach at the end of the word, at the beginning, or maybe somewhere in the middle. Often the affixes give information on which tense a verb has, which case (or number, gender, etc.) a noun has, and a wide range of other things. In Jabberwocky, we could recognize toves as a plural noun simply because we know that the plural form in English very often is formed by adding some ending with an ‘s’ to a word. Thus, we would guess that the root form should be ‘tove’, without having the slightest idea what a ‘tove’ might be for a thingy!
Syntax*

Next in the processing chain is syntactic analysis, or grammar. This is a field which has been of main concern to general linguists and computational linguists alike. What linguists do in order to study languages, is to study the distinction between a speaker’s internalized grammar of his/her language and the speaker’s actual use on different occasions, that is, what Noam Chomsky (an American and one of the leading linguists of this century) called the relation between competence and performance. Chomsky stated that the object of linguistic study should be the language competence, that is, the unconscious knowledge underlying ordinary language use. We could then try to mimic this knowledge by building formal grammars of the languages. This makes sense if we are, as Chomsky likes to claim, born with some models for building grammatical structures “hard-wired” into our brains!

Now, there is grammar and there is Grammar. In high-school, we normally study what is sometimes called prescriptive grammar, that is, try to learn a massive amount of rules by heart. Often these rules aim to make sure that we shouldn’t use certain constructions or forms, such as:

1. “Be sure to never split an infinitive.”
2. “Prepositions are bad to end sentences with.”

Certainly, this can be very useful (and needed) when trying to learn a language... (and certainly, it can be quite boring, too...)

Linguists are seldom concerned with such prescriptive grammar, but rather with trying to determine which aspects of the language use (performance) that reflect properties of the “built-in” grammar (competence). A reasonable way to do this is to ask a large number of native speakers of a language to grade a set of utterances for grammaticality and acceptability — and then try to separate those sentences which are rejected by speakers on grammatical grounds from those which are rejected for performance reasons (that is, because they are impossible for a listener to understand). For example:

3. “It’s raining.” (grammatical and acceptable)
4. “Jill figured that that Tom wanted to take her out annoyed Joe out.” (grammatical but unacceptable)
5. “They are running.” (ungrammatical but acceptable)
6. “Tom slept and the dog.” (ungrammatical and unacceptable)
Thus, the problem of building a grammar for a language can be stated as: “Find a characterization of the processing of linguistic experience (performance) adequate to distinguish unacceptability from ungrammaticality.”

**Representing Grammar**

What the linguists try to do is to formalize this knowledge into particular *grammar rules*. This goes back to another of Chomsky’s claims, namely that the human brain processes language by applying some rules to some kind of abstract representations. These rules then make up the syntactic grammar, since, in addition (the claim goes) the speaker of a language must know the system of rules which is the grammar of that language. Thus, we should in some way be able to reconstruct these rules and write them down using a *grammar formalism* (a language which is used to describe languages themselves).

Now, writing down the grammar can be done in several ways. One option could be to start cutting brains open to find the representation inside them. But this isn’t something researchers like to do anymore (nor do the persons studied like it...). Anyhow, it probably wouldn’t take us that far. Instead, Chomsky suggested studying the performance of the speakers. However, if we are trying to build a Babel fish, this might not be the best way of building our grammar. After all, we know very little about how humans process languages (and even less about how fish do it...!). Maybe we shouldn’t try to copy the human brain’s functionality, after all, but settle for finding a description language which would be efficient for use on a computer.

(As an aside: When scientists started to build chess playing computers, they tried to imitate the way that chess players think. These computer programs barely made it to beginner’s level. Nowadays, the chess playing computers work by doing what computers do best, namely pure number crunching, and play at the level of the best chess grandmasters — but nobody claims that the programs imitate the human thought process anymore. In the same way, there is nothing to say that we even would want to represent and process language on a computer in the same way as a human tackles the problem.)

Thus, most modern computational linguists have left Chomsky’s ideas (neat as they are) aside and concentrated on the task at hand, namely building computationally (rather than psychologically) plausible language representations. One of the main problems to solve then concerns the large *ambiguity* of human language, that sentences normally have several differ-
ent possible interpretations. An analyser (a parser) for a computer language yields a unique interpretation for each input string, while a human language parser must allow for more than one analysis. Consider for example the sentences

(7) “Joe said that Martha expected that it would rain yesterday.”

(8) “She asked him or she persuaded him to leave.”

(9) “He knew the girl left.”

Most humans wouldn’t even notice that sentence (7) has several possible analyses, but try reading it with a slight pause before ‘yesterday’. Was it really the rain which fell yesterday? Or was it Martha’s expectation which took place then? Maybe it even was Joe’s saying so which happened yesterday?!

In the two other examples, some of the ambiguity is actually resolved when we see the whole sentences, but if we read them from left-to-right, even a human encounters some problems. In (9), for example, we could even finish reading after ‘girl’ and have a perfectly ok sentence. But then the word ‘left’ appears and we have to reconstruct our interpretation. (Thus, a sentence like (9) is harder for us to process than say “He knew that the girl left.”)

Semantics*

The sentence in (8) would be rather tricky for the next step in our processing system, the semantic analysis. This process is the one in which we translate the structure built up in the syntactic analysis into an abstract representation of its actual meaning. So, just as for the syntactic grammar formalism, we need to decide on what this semantic representation language is. A lot of work has been done in this area, but it wouldn’t be fair to say that the answer to the best way of representing language meaning is anywhere near to having been found yet. We do know some principles underlying the process, though. A very influential one comes from the 19th century German mathematician and philosopher Gottlob Frege. Basically, he claimed that meaning ultimately flows from the lexicon* and then that meanings of larger groups of words are combined by making use of syntactic information. This is called the Principle of Compositionality (or Frege’s Principle) and is often stated as “The meaning of the whole is a function of the meaning of its parts.”

The effect of this is that we need to be very careful on how we should represent the meaning of the words themselves — in the grammar, we’ll
just compose larger chunks (phrases, sentences, etc.) by building on the interpretation of the actual words. Already the old Greek philosophers tried to represent the meaning of human language by using a formal language, namely predicate logic. It is easy to show that this is far too simplistic, but the general idea still stands; we should try to find a formal language in which we can represent the meaning of our utterances. A main problem, however, comes into play here (well, in fact it comes into play in all the steps shown in the picture...): When talking about what a sentence means, we know that it means something when said in one situation and possibly something completely different in another situation (“Georg is so cool” can mean that George is a great guy... or that he’s freezing... among many things!). To fully understand an utterance, we need to know the circumstances under which it was said, and also have an enormous amount of world knowledge, knowledge about the world which is shared between large groups of people.

Translation and Generation

Too bad, seems like we’ll never get around to building that fish, then, since we’ll spend all our time trying to represent all the knowledge in the world — which we’ll not only need to properly analyse an utterance in one language, but of course also when trying to translate our nifty semantic representation for one language into an equivalent* one for some other language.

Well, things aren’t as bad as they might seem now. If we limit what things people using our fish will be talking about, we only need to represent a portion of the world knowledge in order to do good translations. The device won’t be as useful as we wanted, but still possibly quite useful, at least in certain situations, then. Or we could allow the device to do pretty bad translations, but instead still try to cover all types of situations. (In fact, most present-day machine translation systems go for some compromise between the two goals of being general-purpose or producing high-quality translations).

Having done the translation, we would (hopefully...) have a semantic representation with the same meaning as the original utterance. Using this representation and then building up the sentences to say is the process called generation. In fact, there are two types of generation issues. The first is to decide on what to say, the second on how to say it. The ‘what’-type is often called “deep generation” and is a problem related to planning. The ‘how’-type is called “surface generation” and is what we are concerned with in our fish. In many ways surface generation just reverse the problems of syntactic
and semantic analysis: we have some type of meaning representation and want to get actual words out (words matching the intended meaning, of course!).

Finishing off the Fish

Connecting our generation component to the speech synthesis, we would have a complete Babel fish. Great, we’re done!... Well, there are still a few details to fill in, so I haven’t filed any patent for the fish yet. But you’re welcome to reuse the ideas and build your own fish (just send me 50% of the money you make from it...).2 Certainly having such a device should be very useful, in any case. If people freely could communicate with each other, maybe many conflicts between nations would never surface!

Well, there are different opinions about that too, of course:

“...Meanwhile, the poor Babel fish, by effectively removing all barriers to communication between different races and cultures, has caused more and bloodier wars than anything else in the history of creation.”


Hmmm... so maybe we shouldn’t try to build us a Babel fish after all...

(Still, in the process of looking at the Babel fish, we would have to look at some of the main problems of language and speech processing. Anyhow, the picture on Page 6 actually shows the architecture of a general language processing system, with some functionality that our Babel fish wouldn’t really need to have. For example, whether we would regard the input to the language processing part of the fish as the speech itself or the “written” words already found by the speech recognizer isn’t really that relevant for our present purposes. In general, we could also have a language processing system which was functioning as a kind of preprocessor to an expert system, to some kind of reasoning system, to a relational database, or to some other type of knowledge representation language. Common to all these (and other) types of application would be that the language processing system would output some fairly general representation of a human language utterance.)

---

2But hurry, somebody’s already made an attempt! At least, there is a “Babel fish” service on the net already. Systran, a general-purpose, but low-quality, system with roots back in the 70’s, is available as a translation device through the web pages of Digital Equipment Corporation’s search engine AltaVista™. See http://babelfish.altavista.digital.com.
<table>
<thead>
<tr>
<th>English</th>
<th>Suomi</th>
<th>Svenska</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambiguity</td>
<td>monitulkintaisuus</td>
<td>tvetydighet</td>
</tr>
<tr>
<td>aversion</td>
<td>vastenmielisyys</td>
<td>motvilja, aversion</td>
</tr>
<tr>
<td>chop</td>
<td>hajottaa, paloitella</td>
<td>sönderdela</td>
</tr>
<tr>
<td>chunk</td>
<td>pala</td>
<td>bit</td>
</tr>
<tr>
<td>compare</td>
<td>verrata</td>
<td>jämöra</td>
</tr>
<tr>
<td>computational linguistics</td>
<td>tietokonelingvistiiikka</td>
<td>datorlingvistik</td>
</tr>
<tr>
<td>conscious</td>
<td>tietoinen</td>
<td>medveten</td>
</tr>
<tr>
<td>context</td>
<td>(esiintymis)ymphäristö, konteksti</td>
<td>omgivning, kontext</td>
</tr>
<tr>
<td>device</td>
<td>laite</td>
<td>apparat</td>
</tr>
<tr>
<td>dialogue</td>
<td>keskustelu</td>
<td>konversation, dialog</td>
</tr>
<tr>
<td>equivalent</td>
<td>samanaarvoinen</td>
<td>likvärdig</td>
</tr>
<tr>
<td>induce</td>
<td>päättelä</td>
<td>dra slutsatsen</td>
</tr>
<tr>
<td>intention</td>
<td>tarkoitus, intentio</td>
<td>avsikt, intention</td>
</tr>
<tr>
<td>leech</td>
<td>ilimato</td>
<td>igel</td>
</tr>
<tr>
<td>lexicon</td>
<td>sanasto, leksikko</td>
<td>ordlista, lexikon</td>
</tr>
<tr>
<td>(general) linguistics</td>
<td>yleinen kielitiede, (yleinen) lingvistiiikka</td>
<td>allmän språkvetenskap, (allmän) lingvistik</td>
</tr>
<tr>
<td>manipulate</td>
<td>käsitellä</td>
<td>påverka, manipulera</td>
</tr>
<tr>
<td>native</td>
<td>synnynnäinen</td>
<td>infödd</td>
</tr>
<tr>
<td>nourish</td>
<td>elää</td>
<td>livnär sig</td>
</tr>
<tr>
<td>potential</td>
<td>mahdollisuus</td>
<td>möjlighet</td>
</tr>
<tr>
<td>probability</td>
<td>todennäköisyys</td>
<td>sannolikhet</td>
</tr>
<tr>
<td>semantics</td>
<td>merkitysoppi, semantiiikka</td>
<td>språkinnebörd, semantik</td>
</tr>
<tr>
<td>speech recognition</td>
<td>puheentunnistus</td>
<td>taligenkännning</td>
</tr>
<tr>
<td>speech synthesis</td>
<td>puhesynteesi, puheentuotto</td>
<td>talsynthes, talsammansättning</td>
</tr>
<tr>
<td>syntax</td>
<td>lauseoppi, syntaksi</td>
<td>satsbyggnad, syntax</td>
</tr>
<tr>
<td>tackle</td>
<td>lähestyä, käsitellä</td>
<td>angripa, hantera</td>
</tr>
<tr>
<td>upshot</td>
<td>seuraus, konsekvenssi</td>
<td>resultat, konsekvens</td>
</tr>
<tr>
<td>utterance</td>
<td>ilmaus, lausuma</td>
<td>yttrande</td>
</tr>
<tr>
<td>vast</td>
<td>suunnaton</td>
<td>jätttestor, enorm</td>
</tr>
<tr>
<td>vocabulary</td>
<td>sanavarasto</td>
<td>ordförråd, vokabulär</td>
</tr>
<tr>
<td>warping</td>
<td>vääristävä</td>
<td>(sned-)vridning</td>
</tr>
</tbody>
</table>