An Empirical Approach to Building an Amharic Treebank

Atelach Alemu
Department of Information Science,
Addis Ababa University,
PO Box 1176, Addis Ababa, Ethiopia
atelach@sisa.aau.edu.et

Lars Asker
Department of Computer and Systems Sciences,
Stockholm University and the Royal Institute of Technology,
Forum 100, SE-164 40 Kista, Sweden
asker@dsv.su.se

Gunnar Eriksson
Swedish Institute of Computer Science
Box 1263
SE-164 29 Kista
Sweden
guer@sics.se
and
Department of Linguistics
Stockholm University
S-106 91 Stockholm
Sweden
gunnar@ling.su.se

Amharic is the official government language of Ethiopia and spoken by 15 - 30 million people. In spite of this, it suffers severely from lack of computational linguistic resources. Most work on natural language processing of Amharic to date has consisted of limited prototypes of morphological analyzers, part of speech taggers, and parsers that all suffered from the fact that no lexicon, annotated corpus or tree bank existed. Another weak point of such past work is that most of it was conducted for academic exercise. Once the work achieved their target, no follow-up was made to bring the research work to an operational level or to make the developed resources publicly available.

To overcome this problem, initial steps have been taken to build, and make publicly available, an Amharic corpus, lexicon and tree bank. In this paper we briefly describe this project and its related design issues as well as methodological considerations that are taken to meet the specific needs of the situation. Currently, there is no electronic lexicon or annotated corpus available for Amharic. Furthermore, there are no computational linguists, and

there is hardly any funding available to support the development of a tree bank. On the other hand, there are bilingual (parallel Amharic - English) news (and other) texts available, and a number of relatively good POS taggers and parsers available for English. We are initially working with English and Amharic parallel versions of the Bible that are already available on-line. These texts have the additional advantage that they are already relatively well aligned at the sentence level.

The lack of human as well as financial resources has led us to focus on an example based approach that in addition takes maximal advantage of the resources and computational frameworks that have already been developed for other languages (i.e. English).

Furthermore, in order to automate as much as possible of the otherwise labor intensive and time consuming effort of building a lexicon and tree bank from scratch, we employ a number of relatively simple and straightforward heuristics to support the lexical acquisition. The following is a (simplified) example describing how some of the heuristics can be applied in order to acquire (a draft version of) a partial lexicon:

Abraham begat Isaac .
(1E)
አብርሃም ይስሐቅን ወለደ ::
(1A)

Isaac begat Jacob .
(2E)
ይስሐቅ ያዕቆብን ወለደ ::
(2A)

Jacob begat Judas and his brethren .
(3E)
ያዕቆብ ይሁዳንና ወንድሞቹን ወለደ ::
(3A)

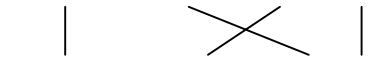
Josias begat Jechonias and his brethren , about the time they were carried away to Babylon:
(4E)
ኢዮስያስ በባቢሎን ምርኮ ጊዜ ፣ ኢኮንያንንና ወንድሞቹን ወለደ ፤
(4A)

In this case, the word "begat" is the only word that occurs in all of the four English sentences. At the same time, the only word that occurs in all four of the parallel Amharic sentences, is the word "**ወለደ**". This indicates that "begat" and "**ወለደ**" are translations of each other and can be added to the lexicon. This conclusion could also have been reached by applying a dependency parser (such as e.g. [1]) to the English sentences to find that

“begat” is a verb (in past tense) and combine that with the knowledge that verbs always appear last in all complete Amharic sentences. In a similar way, it can be concluded (from sentence pairs 1, 2, and 3) that “.” corresponds to “:: “. The word "Isaac" occurs in sentence 1E and 2E, and the only Amharic word that occurs only in sentence 1A and 2A, is "**ይስሐቅ**" (the Amharic letter “**ን**” which comes after “**ይስሐቅ**" in sentence 2A, is a suffix that works as an object marker. The morphological analysis will not be described in this simplified example). Similarly it can be deduced (from sentences 2E, 3E and 2A, 3A) that "Jacob" corresponds to "**ያዕቆብ**". Furthermore, “brethren” (in sentences 3E and 4E) can be linked to “**ወንድሞቹ**” (with the object marker “**ን**” stripped off). The words "Abraham" (in sentence 1E) and "**አብርሃም**" (in sentence 1A) can be linked using the heuristic that they are the only words left (in those sentences), or alternatively, from the relative phonetic similarity between "Abraham" and "Ibrhim" (which is the transliteration of "**አብርሃም**" into SERA[1]). Using the same heuristic, “Judas” (in sentence 3E, can be linked to “**ይሁዳ**” (with the suffixes “**ን**” and “**ና**” stripped off - “**ና**” is a suffix meaning “and”).

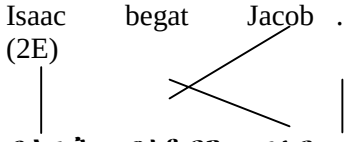
As soon as new lexical items have been determined, they are propagated out in the corpus in order to reduce the search space in the next step. In the example above, we have been able to build up a small partial lexicon, that allows us to make the following assumptions for the first three sentences in the example:

Abraham begat Isaac .
(1E)



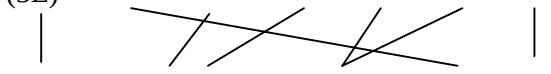
አብርሃም ይስሐቅን ወለደ ::
(1A)

Isaac begat Jacob .
(2E)



ይስሐቅ ያዕቆብን ወለደ ::
(2A)

Jacob begat Judas and his brethren .
(3E)



ያዕቆብ ይሁዳንና ወንድሞቹን ወለደ ::
(3A)

[1] SERA stands for System for Ethiopic Representation in ASCII and is a way to transcribe the Amharic alphabet (Fidel) into Latin characters.

In addition to this, we examine the possibilities to use correspondences between parallel sentences to bootstrap a dependency parser for Amharic. A very strong claim of these correspondences is often referred to as the Direct Correspondence Assumption (DCA), which states that syntactic dependencies hold across languages. In its purest form this is certainly a too strong assumption, but as Hwa et al. [2] shows, some syntactic dependencies, such as head-modifier relations, often hold between parallel sentences in different languages. Our assumption is that this parallelism will make it possible to support the process of disambiguating possible structures for the Amharic sentences, by using the dependency parse of the parallel English sentence.

This approach is then combined with heuristics such as those exemplified above, together with word level alignment using string matching, morphological and contextual information, and linguistic knowledge about specific properties of Amharic (e.g. that the verb always comes last in a sentence) to support the construction of a small tree bank for the part of the corpus that consists of parallel texts. This small tree bank will then be used to train a parser for Amharic that can be applied to the rest of the corpus (that does not have a parallel English version).

It is obvious to the authors that the described approach can not be fully automatic and completely replace the work of human experts. Instead it is intended to function as a semi automatic support to the human experts, that reduces their monotonous workload and instead allows them to focus their attention on the difficult linguistic considerations that have to be taken when building a Treebank.

Until a critical mass of resources such as corpus, lexicon, morphological analyzers and part-of-speech taggers have been made publicly available, it is not likely that natural language processing for Amharic can "take off" and reach a level higher than that of academic prototype systems. The fact that such resources would be made publicly available will further boost activities in the area by allowing researchers to benefit from the work of others and thereby to build more advanced tools without having to invent the wheel over and over again. We believe this to be the best chance for any language with limited computational linguistic resources to break the vicious circle of always having to start from scratch with research in the area of NLP.

References

[1] Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In Proceedings of the 5th Conference on Applied Natural Language Processing, pages 6471, Washington D.C., April 1997. Association for Computational Linguistics.

[2] Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak, ``Evaluating Translational Correspondence using Annotation Projection" 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02), Philadelphia, July, 2002.