Computational Linguistics Group
Department of Computer Science
University of Haifa

בלשנות חישובית
החוג למדעי המחשב
אוניברסיטת חיפה

Laboratory in Natural Language Processing

Shuly Wintner, shuly@cs.haifa.ac.il

Semester A, 2011-12: Monday, 12:00–15:00

1 Objectives

The Lab offers a number of practical projects in Natural Language Processing (NLP), focusing on (but not limited to) processing of Hebrew. Some projects require previous knowledge of computational linguistics but some assume no previous background. All projects involve programming: the end result is a relatively large-scale, well-documented and efficient software package. Some of the projects may involve also some research (e.g., reading a research paper and implementing its ideas).

2 Administration

Projects are to be implemented by groups of at most two students. All systems will be presented at the end of the semester for a final demo. A coordination meeting is planned for Monday, January 30; all work must be completed by the final (presentation) meeting which will be held on Monday, March 5.

The programming language must be portable enough to be usable on a variety of platforms; Python is recommended, C++, Perl or Java will be tolerated, if you have a different language in mind discuss it with the instructor. Some projects may have to be executed in a Linux environment due to dependencies on external packages.

Grading will be based on comprehension of the problem, quality of the implementation and quality of the documentation. In particular, the final grade will be based on:

- Comprehension of the problem (and the accompanying paper(s), where applicable);
- Full implementation of a working solution;
- Presentation of a final working system;
- Comprehensive documentation.

3 List of projects

3.1 A collection of machine translation systems

No prior knowledge is required.

Contemporary machine translations (MT) systems are produced from two types of resources. A parallel source–target corpus of translated sentences is used to extract statistical information reflecting the probability of some source sequence s to be translated to a target sequence t; this is the *translation model*. A large monolingual corpus of text n the target language is used to compute the probability of word sequences in the target language; this is the *language model* .The two statistical models are combined in a system called a *decoder*: when a new source sentence is given, the decoder produces the translation candidate that maximizes some combination of the translation and the language models, thereby optimizing both the faithfulness of the translation to the source input and its fluency in the target language.

Existing resources are available that can generate the two models and implement a generic decoder. Of those, the most popular is Moses (Koehn et al., 2007). Similarly, several large parallel corpora are available for download, such as the Europarl corpus (Koehn, 2005). However, the production of specific MT systems for specific language pairs remains a non-trivial task, which involves some pre-processing of the input, compilation of the best statistical models, system integration, etc. For that reason, very few such systems are available off-the-shelf.

You will implement a set of statistical MT systems from a large number of languages into English, using these available tools. The systems should eventually be freely available for download on the web. You will also implement a web-based demo for these systems. The demo will be used to showcase the systems you build, as well as additional systems involving Hebrew that are produced in the CLG.

3.2 Distinguishing between human and machine translation

Introduction to Computational Linguistics recommended but not mandatory.

Consider the following texts:

> Britain has amended a law that allowed for issuing arrest warrants against Israeli politicians who visit the country, British Ambassador Matthew Gould announced Thursday. Gould called opposition leader Tzipi Livni, against whom an arrest warrant was issued in 2009, and told her the Queen has signed the amendment ”to ensure that the UK’s justice system can no longer be abused for political reasons.”

> British queen has signed today (Thursday) on an amendment to reform the police and social responsibility, to prevent submission of arrest warrants against senior Israeli officials in Britain. Ends legislative amendment process that began following the arrest order was issued against the opposition chairwoman, Tzipi Livni.

Both of them were translated from Hebrew to English; can you tell which one was translated by a human and which one by machine translation?

You will develop a classifier that can distinguish human from machine translated texts. You will be provided with a training corpus consisting of newspaper articles in a single domain in English. The articles will be tagged as either human translated or machine translated. Your main task will be to define a set of distinctive features and implement the feature extractor. Features may include superficial characteristics, such as the average length of sentences or the type/token ratio in a document; n-gram features, such as unigrams of function words, or specific bigrams or trigrams; or more linguistically-informed features, such as n-grams of part-of-speech tags, ratio of active to passive verbs, complexity of syntactic structures, etc. You will be able to use off-the-shelf tools for processing the corpus, and publicly-available machine learning packages for implementing the classifier.

Once the feature extractor is implemented, you will train a classifier on the training material and conduct a robust evaluation of the results. The result of this project will be used in a research on selecting the best language and translation models for machine translation.

3.3 A generic transliteration system

Introduction to Computational Linguistics recommended but not mandatory.

When texts are translated from one language to another, some words are not translated; rather, they are *transliterated*: rendered in the writing system of the target language in a way that retains or approximates the original pronunciation of the word. Transliterated words are frequently proper names or loan words. For example, when the Hebrew sentence 0:3 ספרד הביסה את ברזיל is translated to English, the proper name ספרד is translated to *Spain*, but the proper name ברזיל is transliterated as *Brazil*.

You will develop a generic system for transliterating words in a large number of languages to English, following the methodology of Kirschenbaum and Wintner (2009, 2010). Transliteration will be based on statistical machine translation (Brown et al., 1990), in which the translation model maps characters in the source language to characters in English, and the language model is a unigram English word model (viewed as a character n-gram model). The language model will be provided to you. The translation model will be extracted from multilingual titles of Wikipedia documents.

In order to create a translation model for a given source language, you will have to extract from Wikipedia all the titles of the articles that occur both in the source language and in English, and to determine whether these titles are translations or transliterations. This can be done by comparing the characters in the title terms, given some possible mappings of characters from the source to English. For example, the Hebrew-English mapping will include the pairs ב–*b*, ב–*v*, פ–*p*, פ–*f*, ס–*s*, ר–*r*, ד–*d*, ז–*z*, ל–*l*. Based on such mapping, you will be able to determine that ברזיל–*Brazil* is a transliterated pair, whereas ספרד–*Spain* is not. You will have to prepare such character mapping tables for a few languages.

In order to evaluate the quality of your solution, you will have to prepare an evaluation corpus. This should consist of some 1000 hand-transliterated term-pairs (from various sources). You will evaluate the accuracy of your system on these held-out data.

Variant: a more generic system will allow transliteration to *any* language. Two additional resources will be required:

- a monolingual (target) language model: you will use the monolingual projection of Wikipedia on the target language to create such a language model.
- a mapping of characters between the source and target languages: you will have to provide such mappings for a few language pairs.

3.4 Simplification of Hebrew sentences

Introduction to Computational Linguistics recommended but not mandatory. Real-world sentences can be long and complex. Such complexity is achieved by two main linguistic mechanisms: coordination and subordination. The former allows the conjunction of two simpler sentences, as in: הפלסטינים מבינים שהם בבעיה ואנחנו מנסים להוריד אותם מהעץ יחד עם האמריקאים The latter combines two simpler sentences in an asymmetric way, where one sentence is said to be subordinated to the other: יהיה כמעט בלתי אפשרי למצוא נוסחה שתהיה מקובלת גם על נתניהו וגם על אבו מאזן.

A coordinated structure can in principle be repharsed as two sentences. For example, הפלסטינים מבינים שהם בבעיה ואנחנו מנסים להוריד אותם מהעץ יחד עם האמריקאים can be rephrased as הפלסטינים מבינים שהם בבעיה. אנחנו מנסים להוריד אותם מהעץ יחד עם האמריקאים A subordinated clause can also be simplified by splitting the sentence in two, but this may not be straight-forward. For example, the sentence יהיה כמעט בלתי אפשרי למצוא נוסחה שתהיה מקובלת גם על נתניהו וגם על אבו מאזן can be rephrased as יהיה כמעט בלתי אפשרי למצוא נוסחה. הנוסחה תהיה מקובלת גם על נתניהו וגם על אבו מאזן.

A third type of complexity, frequently observed in journalistic texts, involved quoting. For example, the sentence "אשטון העלתה כמה הצעות ורצינו לשמוע מה הפלסטינים חושבים על זה", אמרו גורמים בלשכת ראש הממשלה can be easily split into "אשטון העלתה כמה הצעות ורצינו לשמוע מה הפלסטינים חושבים על זה". כך אמרו גורמים בלשכת ראש הממשלה

The benefits of sentence simplification are many: such techniques can generate texts that may be easier to understand, for example for language learners. The main motivation of this project, however, is to investigate whether sentence simplification can be useful for improving the quality of an automatic Hebrew to English machine translation system.

You will have to identify linguistic constructions that naturally lend themselves to simplification; stipulate the rules that facilitate splitting one sentence into two or more shorter sentences; and implement a system that used the rules in order to simply arbitrary Hebrew texts.

To evaluate the quality of the system, you will experiment with an existing Hebrew to English MT system, with and without simplification, and compare the results.

3.5 Morphological analysis of dotted Hebrew

Introduction to Computational Linguistics recommended but not mandatory. As you will be revising an existing Java code, knowledge of Java is mandatory.

Morphological analysis is the process of determining the base (also known as *lexeme*, or *lemma*) of a word, along with its morphological attributes. An example of the morphological analysis of a simple Hebrew sentence is depicted in Figure 1.

Hebrew has a complex morphology and hence the design of a morphological analyzer for the language is a complex task. We currently have a large-scale and relatively accurate morphological

```
הרכבת   [+noun][+id]18182[+undotted]הרכבה[+transliterated]hrkbh[+gender]+feminine
        [+number]+singular[+script]+formal[+construct]+true
הרכבת   [+verb][+id]19729[+undotted]הרכיב[+transliterated]hrkib[+root]רכב[+binyan]+Hif'il
        [+person/gender/number]+2p/M/Sg[+script]+formal[+tense]+past
הרכבת   [+verb][+id]19729[+undotted]הרכיב[+transliterated]hrkib[+root]רכב[+binyan]+Hif'il
        [+person/gender/number]+2p/F/Sg[+script]+formal[+tense]+past
הרכבת   [+defArt]ה[+noun][+id]18975[+undotted]רכבת[+transliterated]rkbt[+gender]+feminine
        [+number]+singular[+script]+formal[+construct]+false

שבתה    [+noun][+id]17280[+undotted]שבת[+transliterated]ebt[+gender]+feminine
        [+number]+singular[+script]+formal[+construct]+false[+possessiveSuffix]+3p/F/Sg
שבתה    [+verb][+id]9430[+undotted]שבת[+transliterated]ebt[+root]שבת[+binyan]+Pa'al
        [+person/gender/number]+3p/F/Sg[+script]+formal[+tense]+past
שבתה    [+verb][+id]1541[+undotted]שבה[+transliterated]ebh[+root]שבה[+binyan]+Pa'al
        [+person/gender/number]+3p/F/Sg[+script]+formal[+tense]+past
שבתה    [+subord]ש[+preposition]ב[+noun][+id]19804[+undotted]תה[+transliterated]th
        [+gender]+masculine[+number]+singular[+script]+formal[+construct]+true
שבתה    [+subord]ש[+preposition]ב[+noun][+id]19804[+undotted]תה[+transliterated]th
        [+gender]+masculine[+number]+singular[+script]+formal[+construct]+false
שבתה    [+subord]ש[+preposition]ב[+defArt][+noun][+id]19804[+undotted]תה[+transliterated]th
        [+gender]+masculine[+number]+singular[+script]+formal[+construct]+false
שבתה    [+subord]ש[+noun][+id]19130[+undotted]בתה[+transliterated]bth[+gender]+feminine
        [+number]+singular[+script]+formal[+construct]+false
שבתה    [+subord]ש[+noun][+id]1379[+undotted]בת[+transliterated]bt[+gender]+feminine
        [+number]+singular[+script]+formal[+construct]+false[+possessiveSuffix]+3p/F/Sg

אתמול   [+adverb][+id]12448[+undotted]אתמול[+transliterated]atmwl
```

Figure 1: Example morphological analysis

system for Hebrew (Yona and Wintner, 2008; Itai and Wintner, 2008) which works for *undotted* texts. In this project you will create a variant of the morphological system for the *dotted* script.

The main task here is to understand the morphological rules that apply to words, as stipulated for the undotted case, and then revise and refine them for the dotted case. The greatest benefit of such a system is that it will facilitate, in conjunction with a morphological disambiguation system which is currently under development, an automatic vocalization of undotted texts. Ideally, you should be able to integrate the results of your work with a publicly-available generic speech synthesis system.

3.6 Conversion of transcribed Hebrew to the standard script

Introduction to Computational Linguistics recommended but not mandatory.

CHILDES (MacWhinney, 2000) is an on-line repository of hundreds of corpora recording spoken interactions between children and adults. The Hebrew section of CHILDES contains two large corpora. Both were manually transcribed, and the current transcription reflects both the pronunciation of the words and the specific consonant distinctions of the standard Hebrew orthography. Figure 2 depicts an example; observe that all vowels are reflected, as well as the main stress (as a

horizontal bar over the stressed vowel); observe also that the transcription distinguishes between א and ע, and between ס and ש.

```
*MOT:   bo? nexapēṣ ?et ha- cvaʕīm .
%mor:   v|ba?&root:bw?&ptn:qal&form:imp&pers:2&gen:masc&num:sg=come
        ne#v|xipēṣ&root:xpṣ&ptn:piel&tense:fut&pers:1&gen:unsp&num:pl=search/look_for
        acc|?et det|ha-=the ?|cvaʕīm .
```

Figure 2: Hebrew CHILDES transcription example

We are currently developing a morphological analyzer for the Hebrew CHILDES section, whose output can be seen in Figure 2. One way to evaluate the accuracy of the analyzer is to compare its analyses to the ones produced by the MILA analyzer of *written* Hebrew (Itai and Wintner, 2008). To this end, the transcription has to be converted to the standard Hebrew script.

You will develop software that converts the Hebrew CHILDES transcription to (undotted) Hebrew. You will also develop tools that run the MILA analyzer on the output of your program, and compares the results of the MILA analysis to the morphological annotation available in CHILDES. You will have to develop a set of conversion rules for the two types of analysis. You will also have to overcome difficulties caused by the fact that the CHILDES transcription reflects the vowels, whereas the MILA analyzer assume standard undotted Hebrew. The results of this project will be instrumental for us in improving the CHILDES morphological analyzer.

3.7 A web-based user interface for KWIC in Hebrew

No prior knowledge is required. Understanding of SQL databases and XML is recommended.

Key Word In Context (KWIC) is an algorithm which, given a text and a keyword, presents all the occurrences of the word in the text, allowing a few context words on both sides of the keyword to be displayed. Such a tool is very useful for linguistic research.

You will develop a KWIC system with a web-based graphical user interface which will allow users to present queries referring not just to words, but also to their morphological features. This tool will be similar to an existing GUI for Arabic (Dror et al., 2004), but will be specific to Hebrew corpora. The underlying corpora will be XML documents of morphologically analyzed Hebrew texts. The GUI will enable users to specify a corpus to work with, and then search the corpus for combinations of words and/or their properties. To this end, the corpora will have to be stored in an efficient database; you will be able to use an existing infrastructure for storing corpora, such as The Corpus Workbench. The GUI should be accessible on the Web, and hence will have to be developed in a Web-supporting environment, e.g., JSP or PHP.

A detailed requirements specification will be available in a separate document.

3.8 Extract plain text from Hebrew Wikipedia documents

No prior knowledge is required. Understanding of XML is recommended.

Large textual corpora (collections of texts) are instrumental resources for a variety of natural language processing applications. The largest (freely-available) Hebrew corpus is Wikipedia, with over 123,000 articles containing over 100M words in Hebrew. The Hebrew Wikipedia databased is available for download.

However, before the texts can be used, they must first be extracted from the original Wikipedia documents, which are formatted in XML. Several issues are involved in such extraction: the original documents include images, tables, figures, text in other languages, lists of links, pointers to other languages, bibliography lists, etc. All this extra material has to be removed, and only plain, high-quality text has to be retained.

You will develop a cleanup script that, given an XML document downloaded from Wikipedia, returns only the plain text extracted from the input. You will run your script on a contemporary dump of Hebrew Wikipedia documents, thereby enriching the collection of available Hebrew corpora hundreds of millions of additional words.

4 Available resources

In general, you may freely use any available resources that you find useful for your project (respecting copyright and licensing agreements, of course). Specifically, you may find the following handy:

- Wikipedia as a source of multilingual texts, in particular in order to extract transliterated term-pairs
- Weka, a toolbox of various general-purpose machine learning tools, in particular in order to implement classifiers
- Open NLP, a set of tools for natural language processing, in particular in order to pre-process English texts
- NLTK, a natural language processing toolkit in Python
- The MILA resources for processing Hebrew.

References

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. ISSN 0891-2017.

Yehudit Dror, Dudu Shaharabani, Rafi Talmon, and Shuly Wintner. Morphological analysis of the Qur'an. *Literary and linguistic computing*, 19(4):431–452, 2004.

Alon Itai and Shuly Wintner. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March 2008.

Amit Kirschenbaum and Shuly Wintner. Minimally supervised transliteration for machine translation. In *Proceedings of The 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, April 2009.

Amit Kirschenbaum and Shuly Wintner. A general method for creating a bilingual transliteration dictionary. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 273–276, Valletta, Malta, May 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT. URL `http://mt-archive.info/MTS-2005-Koehn.pdf`.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P07-2045`.

Brian MacWhinney. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition, 2000.

Shlomo Yona and Shuly Wintner. A finite-state morphological grammar of Hebrew. *Natural Language Engineering*, 14(2):173–190, April 2008.