Name: Tomescu Alexandru-Dan

Email: alex.dan.tomescu@gmail.com

Country, city during summer: Romania, Targoviste

Current University and course: Polytechnic University of Bucharest, Computer Science, 2nd year student

IRC nick on freenode: Tomy91

Subscribed to cmusphinx-devel mailing list: yes

Phone number: 0720895878

Provide a link to your CV:

http://www.4shared.com/file/4HGakByF/file.html

Project proposal

Project title: Postprocessing Framework

Description:

Postprocessing Framework refers to a part of the speech recognition process in which the word stream resulted in the basic recognition process is sentence segmented, punctuation is recovered, capitalization is performed and abbreviations are made when needed. This aims to improve legibility and enhance information for future human and machine processing.

Speech segmentation in sentences is an important sub-problem of speech recognition and depends on context, grammar and semantics. This task requires non-trivial techniques, such as statistic decision making.

Spoken language is typically less organized than textual material, making it a challenge to bridge the gap between spoken and written material. The insertion of punctuation marks into spoken texts is a way of approximating such texts, even if a given punctuation mark may assume a slightly different behavior in speech.

The capitalization task consists of rewriting every word with it's proper case depending on context and the abbreviation task consists of rewriting some of the words to a shorter alias (Mister -> Mr).

Reason for choice:

The reason I chose this project is that at a first glance it seemed very interesting, and it has proven so, afterwards. Also, I think a Postprocessing Framework would be a good feature for the CMUSphinx software, as it would improve the resulted text readability, and also I think it would attract more users to discover what the software can do. Also, segmented and punctuated text would be easier to further process.

Automatic punctuation recovery and sentence segmentation is a feature not many other ASR software (or none) have. As stated in [1], this is one of the major limitations of ASR software. In some of the ASR software, the only way to recognize punctuation (like commas or full stops), is if the speaker indicates it through speech. I think automatic punctuation and segmentation would be an important asset.

Dealing with ASR, automatic capitalization provides relevant information for automatic content extraction, name entity recognition and machine translation.

Show us that you've thought about (and/or discussed) what would really be involved in your chosen project

I will discuss the parts of this project by topics. The main topics are **sentence segmentation**, **punctuation recovery**,**automatic capitalization** and **automatic abbreviation.**

Sentence segmentation and punctuation recovery:

As stated in "*Sentence segmentation and punctuation recovery for spoken language translation*", the best results for sentence segmentation (commas and full stops) was achieved for a decision tree that uses multiple features computed for each boundary. The best set of features consisted of word duration of the word preceding the current boundary, pause duration and LM probabilities for comma and full stop insertion.

I believe this should be the way I implement this feature, as studies show it gives the best results.

For the sentence segmentation part, I will need to get as much information on baseline sentence segmentation and decision trees. One possibly effective way to create the decision tree is J.R. Qinlan’s C4.5 induction system.

A large number of punctuation marks can be considered in text: full stops, commas, exclamation mark, question mark, colon, semicolon and quotation marks.

In this project I will consider full stops and commas, as they have higher corpus frequency. The other punctuation marks rarely occur, and are difficult to insert or evaluate.

Automatic capitalization:

Automatic capitalization is treated in [1]. It presents three methods of dealing with the task: (1) an HMM-based tagger, as implemented by the disambig tool from the SRILM toolkit; (2) a transducer, built from a previously created language model (LM); and (3) maximum entropy models.

I have taken into account the first method, which uses a trigram LM and a Map which contains all possibilities of graphical forms of words in the vocabulary. The idea consists of translating a stream of tokens from a vocabulary L (lower-case words) to a corresponding stream of tokens from a vocabulary C (capitalized words), according to a 1-to-many mapping.

But, considering the sentence segmentation and punctuation recovery tasks are complex, in case there's not enough time for this method of implementation, I have thought of another way for capitalization using a database of names. Using the database, the names will be capitalized, and using the sentence segmentation, so will the first word of the sentence. This will be simple to implement, but not as effective, as it will require an ample database.

Of course, if time allows it, or I am instructed by the mentor, I will implement the more complex version.

Automatic abbreviation:

Automatic abbreviation will be implemented using a database with a map of all the words that should be abbreviated, and the abbreviations. Once this step is reached, all words found in the database will be substituted by their abbreviated version.

Research papers on the project you have read (Titles and short resume)

1)"*Sentence segmentation and punctuation recovery for spoken language translation*", Matthias Paulik, Sharath Rao, Ian Lane, Stephan Vogel and Tanja Schultz.

This is the PDF given in the GSoC idea list, and it was the first documentation I read on the matter. It focuses on sentence segmentation and punctuation recovery in the context of Spoken Language Translation (SLT). The paper talks about different methods of processing, on the source (Automatic Speech Recognition - ASR) part or on the target part (Machine Translation - MT), or a mixed method (which had the best results). Two methods of indicating if phrasal context and target LM is jeopardized when segmenting at a given word boundary: using phrasal split-point probability (phrSP) and LM split-point probability (tbiSP - involved in the MT process, which is not actually of interest in this project context).

2)"*Recovering Capitalization and Punctuation Marks for Automatic Speech Recognition: Case Study for Portuguese Broadcast News*", F. Batista, D. Caseiro, N. Mamede, I. Trancoso

This paper treats the recovering of capitalization and punctuation marks. I have focused on the recovery of capitalization in this papers.

The recovery of capitalization is done in three ways, of which the first one caught my attention: using a Map that contains all graphical forms for the words in the vocabulary. This map would be used to capitalize words based on the trigram LM probabilities.

3) ”*Formatting Time-Aligned ASR Transcripts for Readability* ”, Maria Schugrina

What are the goals of your project?

The goals for this project are:

- sentence segmentation which does not affect the meaning of the recognized speech
- punctuation recovery which would help with readability and textual meaning.
- name and first-word capitalization and abbreviation (as accurate as possible)
- overall, I think the main goals are better meaning and text readability as a result of postprocessing.

What is the measure of success for each goal?

- sentence segmentation has to be accurate, as it can modify the whole meaning and ruin the purpose of the recognized speech.
- punctuation recovery should be as accurate as possible, and not change the intended meaning of speech.
- capitalization and abbreviation should also be as accurate as possible and they should improve text readability.

Milestones (at least 3)

- the first and most important milestone I think is sentence segmentation. I believe this part will be the hardest to tackle.
- the second one will be the punctuation. I will do this using English baseline speech segmentation
- automatic capitalization will be the third milestone. This step will greatly benefit from sentence segmentation, as the first word capitalization will be based on it.
- automatic abbreviations is the last milestone.

What is your planning schedule for completing these goals? (preliminary, for further discussion)

- Until the main coding phase starts, I plan to document on how to approach the project in the most effective way, and to get used to the program, maybe help with some bugs (I think this is the best way to get used to software programs and it would help the CMUSphinx community). By the time I start the coding phase I should know exactly how I would implement the framework features.
- The sentence segmentation should be the first goal. It should be done a week before mid-term in July.
- Punctuation should be ready in about two, three weeks at most.
- Word capitalization should not take more than a week and neither will the abbreviations.
- The remaining time until the deadline I will solve the code bugs and try to improve my work.

What are your plans after the project

After completing this project, I plan on maybe contributing with a follow-up project in postprocessing, to help build a more complete framework.

If this will prove to be enough postprocessing I will move on to other projects to help develop the framework (maybe the web data collection crawler, if no one does it by then).

Duties:

We expect you to work on the project 30 hours per week. Are you ready for that?

Yes, I am ready the work full time on the project. 30 hours per week will probably be the minimum. I plan on working around 8 hours per day (including some weekends). In the summer of 2011 I worked full time as a web developer, with 8 hours or more per day, so I know I will be able to handle it.

Do you have any commitments during the summer?

I have no commitments this summer.

Exams or other events you expect to have to deal with during the GSOC period.

My exam session starts on 19th of May and ends on 8th of June, and the GSoC timeline says coding starts on 22nd of May, which means for 2 weeks I won't be able to concentrate on GSoC. If it's possible I would like to start coding earlier or try to make up for the 2 weeks after my exams. I will try to keep in touch even during my exam session, but I won't be able to do a lot, as I have some challenging courses this semester.

How you plan to juggle the competing demands on your time

I plan to be as efficient as I can, but as I said, doing overtime won't be a problem. It's very important for me to finish what I start.

Note that we require a minimum of weekly contact from all our students, unless forewarned

That won't be a problem. I plan on being online pretty much everyday on IRC.

We expect you to blog about project success each week. Are you ready for that?

Sure. I will provide weekly updates on my work.

Will you have an Internet access during the summer

Yes

Experience

Programming languages you have learnt, and how many lines of code, approximately, you have written in each.

C - I started in high-school, and really learnt alot in the first year of college. I'm familiar with data structures (I believe this will be very useful in implementing the project I have chosen). It's hard to estimate a number of lines, but I would say about 5000 lines.

Java - Around 1000 lines. I am familiar to Object Oriented Programming and I find it very useful.

Python - Started learning python a few months ago, and I am not very experienced, but I find it pretty easy and intuitive. This is one of my goals for the summer: to get used to python. I think I wrote about 300 lines of code, while doing some tutorials and exercises.

PHP/HTML/CSS/MySQL - In the summer of 2011 I worked full time as a web developer (I worked on the development of Content Management Systems), and I estimate I wrote about 3000 lines of code. I am also familiar with networking topics as I am enrolled as a student for CISCO classes.

C# - I participated in a competition for two years in a row with some projects. I would say i wrote about 600-700 lines.

Have you ever involved in scientific research? Do you read scientific papers?

No. I haven't been involved in scientific research until now, but I hope that this summer (and from then on) I will be involved in such activities. This is another goal for the summer, and a reason I want to get involved in CMUSphinx: to be a part of research I find very interesting.

Describe your math experience

Math has been a very important part of the education I was given. I learned the basic things in high-school, and in college I had a lot of math courses in the first year. This second year I got to use math in other courses. I am familiar with Fourier and Laplace Transform, and other notions and I am familiar with Matlab. Even if the project requires something I haven't yet studied, it won't be a problem.

Describe your machine learning experience

I have no experience with machine learning. I've had a bit of contact with artificial intelligence while working on a project for a university course: a version of Google's AI Challenge contest of 2012 (Ants).

Up and running

Have you succeeded pocketsphinx from subversion

Yes. I haven't worked with subversion before, but I have experience with Git.

Provide a link to the log of pocketsphinx speech recognition session on your computer (THIS IS A STRONG REQUIREMENT)

TODO (Eventually done)

Open source development experience

Is this your first contact with the CMUSphinx project?

Yes.

List or link to any code, patches, or bug reports contributed to other projects

I haven't contributed to any other open source projects.

List or link to any code, patches, bug reports contributed to the CMUSphinx project

I have submitted a patch to cmusphinx for SphinxTrain VTLN training. I moved the warp factor feature extraction to the 000.comp_feat stage:

https://sourceforge.net/tracker/?func=detail&aid=3515214&group_id=1904&atid=301904

Why CMUSphinx

Before the GSoC projects were announced I hadn't really thought about Speech Recognition. It was an interesting topic (especially with the advance of mobile systems and many field of application) , but I never got to read anything about it.

When I started trimming the list, 2 or 3 projects got my attention, and after thinking about it, I chose CMUSphinx. It's my best chance to get involved into something very interesting this summer, and get involved into the open source community. As I've worked and looked through the code and the functionality I am now very motivated to give it my best.

I hope that I will get to work on this project as part of GSoC 2012.

[1] - http://autocaption.com/tools_caption_speechrecognition.html

[2] - http://www.cs.cmu.edu/~ianlane/pub/PAULIK-icassp08.pdf

[3] - http://peer.ccsd.cnrs.fr/docs/00/49/92/19/PDF/PEER_stage2_10.1016%252Fj.specom.2008.05.008.pdf