GSoC'16 Proposal

Develop a spell checking system

Ashutosh Ranjan
ashutosh.ranjan019@gmail.com
IRC : **ranjan19**
+917207222154
India (UTC+5:30)
Link to Apertium wiki Profile (Contributions)
Link to GitHub Profile (Not show-worthy, but yes it exists!)

About me and Why am I interested in Machine Translation?

I am a second year student pursuing **BTech in Computer Science** and **MS by Research in Computational Linguistics** at International Institute of Information Technology, Hyderabad.

Our college focuses on research at the undergraduate level (3rd year onwards) and so my research work would start this summer. The above mentioned degree which generally takes 6-7 years (4 + 2-3) is shortened to 5 years in our college majorly because of the facts that **(1)** we start our research right after 2nd year, **(2)** along with the core Computer Science courses we are required to do courses in Computational Linguistics from the very first semester itself. So, having spent almost four semesters here I have acquired a good amount of basic knowledge in Linguistics.

This instills my interest in Machine Translation. The first time I was awed by linguistics after coming to college was when I was supposed to do a project where one of basic tasks was to translate 50 sentences from Hindi to English, two languages I have been using since birth. The most enlightening thing about this was that it was not at all easy to translate (realising that there was no word-to-word mapping for almost any of the sentences as I was expecting), which I took for granted before. Since then **we have studied in detail various intricacies of languages, speech and trying to find ways in which we could make a computer understand a language and its different aspects**. **This is where machine translation comes in, and my love for it.**

Why am I interested in Apertium?

GSoC is a very popular summer project aspired for by the students of my college with a lot of people having done it in the past few years (stats). The above mentioned factor plays an

important role in getting new students (1st years and 2nd years) motivated and drawn to **open source**. I came to know about Apertium from one of my seniors who has worked here as a GSoC student before and still does some contributions from time to time (if I'm not wrong! :D). Also since I am a linguistics student (and **Apertium being linguistics oriented**) I started up with this organisation in around December end last year (with a different IRCnick: artss264). I made the hin-eng language pair work in my system then and that was when I understood part of how Apertium works. **The main IRC channel of Apertium is very active and that's what I like the most about this organisation**. This helps a lot and has helped me a lot in the past in understanding the project and completing the coding challenges under the given project.
So answering the question "Why am I interested in the Apertium project?," I would say that this is the first organisation I came to for open-source experience and for GSoC purposes and I never had to look back or go to any other organisation for that matter. Having a linguistics background helps.

Now that I have started open source (to be honest, in the name of GSoC) I find it very pleasing and wish to continue being a part of it. **It really boosts my confidence when I am talking to awesome and extremely knowledgable people from around the world, learning from them and actually making a difference.**

Project idea I am interested in: Develop a Spell-Checking System

Abstract

The aim of this project is to make it easier to **make spell checkers from Apertium's transducers** of all different languages, **and integrate them into the web site**. The transducers can easily be used as spellcheckers by compiling them to ATT files which in turn can be converted to HFST format (if not already in the HFST format). The HFST format can be converted to ZHFST which is a library that can be used as a spell-checker by hfst-ospell or voikkospell. Many languages based on HFST in the languages module of our repository have this set up, but not all of them. So here the objective would be to make spell-checkers available for all languages of all build types and implement spell-checking as a feature (mode) on the apertium website.

Why Google and Apertium should sponsor it?

Spell checking is one of more useful services apertium's language resources can provide. Being able to easily provide spell checking is important both for language communities to be able to use the resources and as a new entry point for prospective developers.

Project Plan

This project would primarily be divided in three main parts:

1. **Making compiled spell-checkers available for monolingual modules (back-end)**

The first three tasks given on the spell-checker project page come under the purview of this part of the project, which are-

- Both hfst and lttoolbox transducers can be compiled to zhfst spellers, dynamically generating the error model where none exists.
- Create clean Makefile rules for speller compilation that are usable in apertium's monolingual modules.
- Integrate spell-checking into apertium as a configurable option to compile.

Apertium **hfst transducers can be compiled into libraries that libvoikko (voikkospell) or hfst-ospell can use to perform spell checking**, including providing suggestions. The lttoolbox based transducers can be converted to hfst and then to zhfst and then be used in the same way. The tasks mentioned above would be done with comprehensive **incremental testing**.
After accomplishing the above goal the focus would shift to determining the best way to make **spell-checking integrated into apertium** as a configurable option to compile.

2. **Completing the spell-checking web interface (front-end)**

The last two tasks given on the spell-checker project page come under the purview of this part of the project, which are-

- Finish the web interface for apertium-html-tools for spell checking and make it interface with APY.
- Make Apertium-apy support spell checking as an apertium mode i.e. integrate spell-checking into apertium as an apertium mode.

This part of the project is concerned with the development of a **spell-checking interface for apertium**. This would include finishing the web interface for apertium-html-tools for spell checking making Apertium-apy support the above mentioned type of mode. The prototype UI for spell checking is already in motion, I would follow up on that.
Options for interface design (subject to discussion and change)-

1. A prototype (link) for the spell-checker interface already exists. It underlines the words but has no **back-end support**. I would be improving the interface (there's a lot of scope), fix the bugs, make it user friendly and give it the required backend support where a "right-click" or a "hover" over the wrong (underlined) word would give correct suggestions. If this option is taken the spell-checking interface would be one of the "**modes**" on the main apertium webpage. The existing modes right now are **translation**, **morphological analysis** and **morphological generation**.

2. The other option could be - The main website of apertium which right now serves the purpose of translation could also serve as a spell-checker whenever a user inputs a wrong word while trying to translate. The wrong word would be underlined and a "hover" or "right-click" would give user correct options to select from (**back-end implemetation**). The feature of "underlining" should be easy to implement as the textbox on the main webpage already does the task of identifying wrong words (uses asterisk). This asterisk could be replaced by "underlining" in the source code pretty easily. This option if taken would **relate translation and spell-checking**. This is probably more in line with what people are used to.

One of the the major challenges could be making the system realise errors in multiwords, for example (pointed by Unhammer) "**to day**" is a multi-word which should provide a suggestion "**today**".

This part of the project would take a major portion (supposedly two-thirds) of the three months I am supposed to work on this project.

3. Testing and Documentation

I would be spending some time(see timeline) testing the **first couple of languages** (of each build type- hfst, lttoolbox and others if any) I implement the makefiles on, made during part one of the project. If it works well for a couple of languages it would work well for the others unless the dictionaries of those languages are faulty. This would be done before Mid term evaluation 1. Would do the same for the web interface when it gets ready at the end of the project.

This would mark the end of the project where I would **test** the web interface comprehensively and **once we are done with this part we would be able to do web-based spell checking smoothly and effectively**. I would make sure the spell checking interface becomes usable as a web service/ apertium mode at the end of the project.
I would also focus on complete and thorough **documentation** of the project through the course of the project and at the end where a couple of weeks have been reserved for testing, bug fixing and documentation purposes.

This was the minimalistic workflow cum description of the various parts of my project. The timeline (coming next) describes the schedule that I would be following for the project.

Timeline

Weeks (1 Week = 40-50 hours)	Plan
[23nd Apr-22nd May] (Pre-Project)	Community bonding. Getting to know more about how apertium-init, apertium-html-tools and apertium-APY work. Also learning about hfst to zhfst conversion.
[23rd May- 29th May] (1 Week)	Compiling hfst and lttoolbox transducers to zhfst spellers, dynamically generating the error model where none exists.
[30th May- 5th June] (1 Week)	Create clean Makefile rules using the information learnt in Week-1 for speller compilation that are usable in the monolingual modules.
[6th June- 7th June] (2 Days)	Testing the above makefiles on differently built languages to confirm their accuracy.
[8th June- 14th June] (1 Week)	Make spell-checking integrated into apertium, as a configurable option to compile. (integrating this into apertium-init)
[15th June-19th June] (5 Days)	Running Tests for the above accomplished tasks, documenting and submitting as Deliverable 1. (Mid Term Evaluation)
[20th June- 6th July] ($2_{1/2}$ Weeks)	Start up on the web interface. Complete most part of it.

[7th July- 14th July]	**Would be going home. Would stay in touch but would not be able to give more than 10-15 hours.**
[15th July- 31st July] ($2_{1/2}$ Weeks)	**Finish the web interface for apertium-html-tools for spell checking. Integrate into apertium website as an apertium mode.**
[1st August- 10th August] ($1_{1/2}$ Weeks)	**Make it interface with APY. Make Apertium-apy support this type of mode. Make the interface live-use-ready.**
[11th August-23rd August] (2 Weeks)	**Bug fixing, testing and documentation.**
[24th August]	**End of Project**

I would have summer break during the project duration and have no prior commitments so would be able to devote most of my time(6-8 hours 7 days a week) to the project.

Also apart from the above mentioned ideas, in future I would like to add more features to this project if need be, and work on other different projects while carrying on as a regular contributor to Apertium.
You see, I have only started as a Computational Linguist. I have a lot to give :).

Contributions to Apertium and Why me?

If you ask me what is the **best quality** I have, I would say **reliability. I am extremely reliable.**

But perhaps that's not the entire expectation of this question. Moving forward to my **contributions** **in Apertium**, I have worked on the **first two coding challenges** related to the project spell checker system.

1. **Compile a speller for one of our hfst-based analysers and run it on some text**

I did this **coding challenge** with the help of this wiki page. Having a wiki page already available made the challenge pretty easy. But there was an error on this page that was not allowing me to complete the challenge in one go. **I found the error and fixed it myself (my first fix :))**. It was a minor fix but it could have annoyed a few people before being sorted out. The **fix** kind of speaks for the fact that I completed the coding challenge.

2. **Compile a speller for one of our lttoolbox-based analysers and run it on some text**

This **coding challenge** took a lot of time and patience. It is also a part of a task of the project. There were two parts to it **(1) Converting a lttoolbox binary into hfst** and **(2) Converting hfst to zhfst to be used as spell-checker with hfst-ospell or voikkospell**. I completed the first part successfully but even after a lot of discussions and effort the second part is yet to be completely resolved even though I have moved very closer to the solution. **I have made a wiki page that documents the whole process and also mentions the error.** The current page after a few changes is this. The page would be moved as and when the issue is resolved.

During both of these discussions I was in constant touch with the mentors on the IRC channel.

Projects and Academics

In the last two years since I have joined college I have worked on various projects that were done (and I'm **proficient in**) mostly in **C, Python, html, css and angularjs (not so much)**. I also have a fair amount of **bash scripting experience** of which I personally am a big fan. These links mentioned below are some of my **project works**.

1. AI Ultimate-Tic-Tac-Toe bot/player(python)
2. Graphics Shooting Game(like Angry Birds, C/C++)
3. Graphics 3D game(maze/grid, C/C++)
4. Library portal(Web2py framework, python)
5. Android chat bot to book flights(js, angularjs, html, css)

Linguistics' Presentations

1. Generating Monolingual Dictionary using Corpus based Paradigm Matching
2. Turn-Taking analysis: FRIENDS TV SHOW

Apart from these I have done courses in **Data Structures, Algorithms, Operating Systems, Software Engineering principles, Digital Logic and Processing** and **Computer System Organisation**. Also I would be completing courses like **Computer Networks, Graphics, Formal Methods (Regular/Irregular languages, DFA, NFA)** and **Artificial Intelligence** this semester so I have a good knowledge base in all of these.
In **linguistics** I have done basic courses in **phonology, morphology, sociolinguistics, semantics, pragmatics, discourse** etc. During these courses I have worked on **IUPAC transcription, phrase structures, dependency trees, event semantics, conversational and discourse analysis and a particular project on creating a monolingual self-improving dictionary for Bangla language**

using corpus based paradigm matching (link added above). I also presented an analysis of "Turn-taking in the TV show Friends" (link added above) a few days ago.

IRC Activity (IRCnick: ranjan19)
I have been **fairly active on the Apertium IRC channel** for the past 3 weeks. I know it's not a very significant thing, but yes I can boast about it :D.
Last-28-day-stats (6th March-Today) | Also see the most referenced nicks, most words
7-day-stats (6th March- 12th March)

I have worked very hard on this project in the last 3 weeks and have understood the project very well. I believe I would be able to complete the project in the given time, if not earlier and would exceed your expectations. I understand that my github profile is not that glossy and well built but I'm still a newbie and I'm getting better.

I would love to have a chance to work on this project for the summer, it would be life-changing for me, literally (strictly NOT an exaggeration)**.**