Software Project

Artem Sokolov
Computerlinguistik
Universität Heidelberg
Sommersemester 2015

(includes material by Laura Jehl)

1 Overview
2 Projects Presentation
- **Audience:** bachelor students
- **Requirements:**
  - Programmierprüfung
  - participation in Ressourcenvorkurs
- **Duration:** 1 semester
- **6 LP + 4 LP ÜK**
1 Overview

2 Projects Presentation
You will have to accomplish an NLP/ML task working in autonomous teams:

1. planning
2. implementing
3. testing
4. documenting
5. packaging
6. presenting
You will learn how to:

1. develop a practical plan from an abstract task
2. map the plan to viable work-packages and team-member assignments
3. present and analyze your results to other teams
You will learn how to:

1. develop a practical plan from an abstract task
   - reformulate the task in your own words
   - modularize the task, define inter-relations and reusable parts
   - split responsibilities among your team fellows
   - prioritize the sub-tasks and develop a project timeline
   - define architecture, data structures and interfaces

2. map the plan to viable work-packages and team-member assignments

3. present and analyze your results to other teams
You will learn how to:

1. develop a practical plan from an abstract task
2. map the plan to viable work-packages and team-member assignments
   - design common libraries and routines
   - timely implement work-packages assigned to you
   - document and test your modules for other members to use
   - settle on means of communication and issue tracking, ensuring no organizational blocks arise
3. present and analyze your results to other teams
You will learn how to:

1. develop a practical plan from an abstract task
2. map the plan to viable work-packages and team-member assignments
3. present and analyze your results to other teams
   - clearly present the project’s goal
   - explain the main difficulties and problems you encountered
   - demonstrate the results and explain them
   - argue further improvements
- regular meetings in a team (in person or video) and with me
- shared logbooks: wiki/notes/documents
- shared calendar
- weekly status reports of all team members
- code-reviews
- refactoring
Dates, events, announcements:

http://www.cl.uni-heidelberg.de/courses/ss15/softwareprojekt
Send before Friday, **24.04** to sokolov@cl.. an email with:

1. **subject**: “SWP Anmeldung”
2. projects you’d like to take in decreasing order of priority (rate all 6)
3. programming languages you know with a grade
   (1: no experience,..., 5: mother tongue)
4. names of team mates
   (if you already know with whom you’d wish to team up)
<table>
<thead>
<tr>
<th>date</th>
<th>week</th>
<th>event</th>
<th>details</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.04</td>
<td>1</td>
<td>intro today</td>
<td></td>
</tr>
<tr>
<td>28.04</td>
<td>2</td>
<td>team formation</td>
<td>next week</td>
</tr>
<tr>
<td>05.05</td>
<td>3</td>
<td>kickoff meetings</td>
<td>bring questions!</td>
</tr>
<tr>
<td>12.05</td>
<td>4</td>
<td>plan</td>
<td>2 pages, by email</td>
</tr>
<tr>
<td>19.05</td>
<td>5</td>
<td>status report</td>
<td>per team, in person</td>
</tr>
<tr>
<td>26.05</td>
<td>6</td>
<td>specification</td>
<td>slides</td>
</tr>
<tr>
<td>02.06–14.07</td>
<td>10</td>
<td>status reports</td>
<td>7 weeks per team</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>in person/by email</td>
</tr>
<tr>
<td>21.07</td>
<td>14</td>
<td>presentation</td>
<td>slides, last meeting</td>
</tr>
<tr>
<td>28.07</td>
<td>15</td>
<td>delivery</td>
<td>package, <strong>hard deadline</strong></td>
</tr>
</tbody>
</table>
Splitting into Teams – 28.04, 14:15

- 3..5-member teams
- tentative project assignments
- splitting into teams (teams can be preformed beforehand)
- everyone gets a project
Kickoff Meetings in Teams – 05.05

- read and discuss articles
- identify questions or unclear things
- elaborate plan (tasks, timeline, responsibilities)
Project Plan – to be sent on 12.05

1. description of the task (goal)
2. ... and solution plan (method) in your own words
3. evaluation methods/measures
4. what datasets and tools are necessary

- write-up: ~2 pages
First Status Report – 19.05

1. status of the specification preparation
2. can do by email or in person
Specification Report – to be handed on 26.05

1. content:
   - task, solving approach, evaluation (from the *Plan*)
   - selection of required resources and methods/algorithms

2. modularization & task distribution:
   - definitions of modules/tasks
   - sub-task assignments to people
   - timeline

3. concrete planning:
   - program architecture
   - data structures
   - programming languages

- oral report with slides: 20 min. + 5 min. questions
Status reports: 02.06 - 14.07

- on individual basis with every team
- normally – in person, weekly (Tu, 14h-18h)
- reserve a 30-45 min. slot from 14h to 18h, by email
- no meetings in other time
- if all goes as planned – send accomplished milestones by email
  - still plan at least a bi-weekly meeting in person
  - 30.06 – all teams report by email
Final presentation – 21.07

1. your implementation of the task and the approach
2. presentation of evaluation results
3. “demonstration” of the running system
4. lessons learned:
   - what hypothesis proved to be wrong?
   - what new hypothesis were called to replace them?
   - what did not work?
   - what could be done better?

- slides, max 25 min. + 10 min. questions
- optional: poster
Final deadline for a packaged project – 28.07

1. this is hard deadline
2. delivery by email/link
3. “should just work”

Requirements for passing:

- participation/reporting on all dates
- plan
- specification
- final presentation
- documentation + packaging
  - documentation of source text
  - README / INSTALL
1 Resources:

https://wiki.cl.uni-heidelberg.de/foswiki/bin/view/Main/Resources/webhome

2 Style guide:

https://wiki.cl.uni-heidelberg.de/foswiki/bin/view/Main/Resources/SoftwareStyleguide

3 recommended: svn or git
1 Overview

2 Projects Presentation
## Projects Overview

<table>
<thead>
<tr>
<th>Software Project</th>
<th>MachLearn</th>
<th>DeepLearn</th>
<th>CrossLang</th>
<th>InfRetr</th>
<th>(Big)Data</th>
<th>HumanComplInt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1 Motivation (why?)
2 Idea (how?)
3 Task (have to do this)
4 Info (recommended data, tools)
5 Research Problems (need a challenge?)
Corpus Harvesting from Quasi-parallel Linked Data

Motivation:

- how to induce knowledge from linked data?
- medical research uses terminology inaccessible to general audience
  - although only one language is used
- example: linking observed symptoms to a preliminary diagnosis
- quasi-crosslingual information retrieval problem:
  - layman English $\rightarrow$ “medicalese” English
  - “brain worm” $\rightarrow$ “neurocysticercosis”
need to map terms from everyday to terminology-heavy English.

learn on examples of common English and relevant medical articles

how do we know about the relevance?

from citations that documents make to scientific research!
ALS (Lou Gehrig's Disease): Fishing for Answers

The neurotoxin BMAA is found in seafood and the brains of Alzheimer's and ALS victims. Might dietary changes help prevent amyotrophic lateral sclerosis?

Latest Health & Nutrition Videos

Nutritional Yeast to Prevent the Common Cold
Beta glucan fiber in nutritional yeast may improve immune function but there is...

Can Oatmeal Help Fatty Liver Disease?
Is whole grain consumption just a marker for healthier behaviors or do whole...

Oatmeal Lotion for Chemotherapy-Induced Rash
Oats are put to the test against cetuximab-type chemotherapy side effects to see just...
1. create a large train/dev/test dataset
   (http://nutritionfacts.org)
   - crawl transcripts of videos, description, notes and summary articles
     (plain English)
   - crawl and extract text from linked research pdf articles
     ("medicalese" English)

2. come up with a meaningful graded-relevance scheme, e.g.
   - direct link - 1, linked from a linked article - 2, same category - 3...
   - consider links from additional fields: comments, doctor notes, descriptions

3. choose some (CL)IR baselines, e.g.:
   - SMT + IR engine: e.g. cdec + Lucene, Moses + Lucene
   - standard metrics: tf-idf, bm25
   - approaches from our group
Recommended Tools

- Python (crawling: sitescraper, pdf: pypdf/pdfminer)
- unix tools (pdftotext, libpoppller)
- IR engines (Lucene/Xapian)
- hadoop or multi-threaded evaluation
- look at in-house tools from our group

Info

- for people with interest in Web, CLIR and SMT
- Data: to be created in the project
- Evaluation metrics: MAP/NDCG/PRES
- Literature for ideas: [Schamoni et al., 2014, Sokolov et al., 2013]
- try query expansion with UMLS [Eck et al., 2004]
- improve over CLIR baselines
- learn an SMT system from quasi-parallel data
Feedback-based Learning for Machine Translation

- obtaining strictly parallel data is time-consuming & expensive
- weak binary feedback ("ok"/"garbage") is relatively cheap/easy
  - is possible to get even from the users who are monolingual
- this kind of feedback is a natural fit to online SMT applications (smartphones)
  - where quick translation improvement is highly desirable and
  - repetitions of the same mistake annoy users
incorporate the binary feedback into the max-margin structural loss of the SVM

the loss will combine both

- the fully supervised sub-dataset
  (where references exist)
- online weekly supervised sub-dataset
  (where only binary feedback is available)

\[
\begin{align*}
\min_w \frac{\lambda}{2} \|w\|^2 + & C_1 \sum_{i \in S} \left( \max_y [\Delta(y_i, y) - w \cdot \Phi_{y_i, y}(x)] \right) \\
+ & C_2 \sum_{i \in B} \max \left( 0, 1 - \ell_i \max_h w \cdot \Phi_B(x, \hat{y}_i, h) \right)
\end{align*}
\]
1. implement the approach of [Saluja and Zhang, 2014]
2. evaluate on FR-EN and PT-EN datasets
Recommended Tools
- decoders: moses (C++) or cdec (C++/Python)
  - study existing implementation for cdec: http://github.com/asaluja/cdec

Info
- for people with interest in SMT, human-computer interaction, commercial NLP
- Data
  - PT-EN part of the Unbabel corpus (have it in-house)
- tackle the problem of long sentences
- design a better update
- improve over baselines
Send before Friday, 24.04 to sokolov@cl.. an email with:

1. subject: “SWP Anmeldung”
2. projects you’d like to take in decreasing order of priority (rate all 6)
3. programming languages you know with an experience grade (1: no experience,..., 5: mother tongue)
4. optional: names of team mates (if you already know with whom you’d wish to team)

