School of Languages, Linguistics and Film Assessed Coursework Coversheet

For undergraduate (BA) modules coded:
CAT-, COM-, EAL-, FLM-, FRE-, GER-, HSP-, LAN-, LIN-, POR-, RUS-, SML-

Please read and note the following guidelines:

1. To assist with anonymous marking, please use your nine-digit student ID number only: do **NOT** use your name anywhere on your coursework.
2. Normally you will be required to submit one electronic copy of coursework via the module's QMplus area. Most deadlines in this School are set for a Sunday night (23:55). You will be informed by the module organiser of any exceptions to this procedure, either regarding the time or method of submission. It is your responsibility to ensure that you know and meet the submission requirements for each piece of coursework.
3. You must keep a copy of all coursework you have submitted.

4. Extensions to deadlines may ONLY be granted by the Senior Tutor for your department. In order to be granted an extension, you must submit a claim for Extenuating Circumstances BEFORE the coursework deadline. SLLF has an online EC claim form. Details and links to the form can be found on QMplus School of Languages, Linguistics and Film Landing Page.

5. Late submission, without an agreed extension due to extenuating circumstances, will be penalised according to the QMUL regulations relevant to your level of study.
6. Work submitted within 7 DAYS of the deadline will be accepted but subject to a late submission penalty against the marks awarded. The work will be marked normally, and then a late submission penalty of five marks (or 5% of the marks if not marked out of 100) per 24 hour period will then be applied.
7. Work that is more than 7 DAYS late will not be accepted and will not be marked and will receive a mark of ZERO.

You are reminded that plagiarism, that is copying someone else's words or ideas without attributing them to that person, is cheating. This is a serious examination offence and at the very least will result in a mark of zero being awarded for this piece of work; it could result in your expulsion from Queen Mary.

By handing in this coursework you acknowledge that it represents your own, unaided work and that you have appropriately acknowledged all sources.

Please complete the following details:

Student ID Number:(9-digit number): 190245566
Module CODE and TITLE: LIN6209 Coding for Linguists
Title of Coursework: Week 8 Assignment Part 1: Project Proposal
Essay no:
Number of words written: 577
Module Organiser: Peter McGinty
Seminar Tutor (if applicable):

Please continue your coursework on the next page

Classifying Smiled and Non-Smiled Speech using Acoustic Measurements and Machine Learning

This project aims to build a classifier that can categorize acoustic measurements as belonging to smiled or non-smiled speech.

Humans are very good at identifying when their interlocutor is smiling, even without any visual cues (ref). Smiling affects the articulatory system, which means that smiled speech (SS) has different acoustic features to non-smiled speech (NSS). Previous research has measured these differences in speech production and found SS usually has higher pitch and formant frequencies (ref). It is still unclear which of these features are most affected by smiling and if any of them are particularly important for perceiving and correctly identifying SS (ref).

Moreover, previous research has focused on directly comparing instances of the same sounds in SS and NSS by the same speaker (ref, ref, ref). This means that their results would only apply to pairs of utterances and not to single ones. However, humans show no added difficulty in identifying SS in interlocutors they have never heard before (ref).

This project will aim to build a Python program that can take a single set of acoustic measurements belonging to the articulation of a vowel as its input, and classify it as either SS or NSS. This will be done use Supervised Machine Learning tools and libraries such as Scikit-Learn.

Further research stemming from this project can carry out an analysis of the resulting sorting algorithm which might shed light on which features are more affected overall by smiling. This program could also be modified to identify any other phenomenon that might affect the articulatory system and integrated in automatic acoustic measurement software and speech recognition software to aid in sentiment analysis.

The data for this project will come from the measurements previous researchers have taken when investigating SS. They will have already analyzed and classified sounds, and will be able to provide files (txt or csv) with measurements. I will likely require data from multiple previous projects, but the owners of the data have shown interest in collaborating for this project and I will be able to obtain enough data to train the model. Due to the diverse sources, the first step of this project will be to collect, clean and sort all data in a database. I expect the following variables to be relevant: target vowel sound (noted in AA), the following sound (voiced or unvoiced), F0, F1, F2 and F3 (measured in Hz), and the categorical distinction between SS and NSS. The resulting database will look like this:

record	target_sound	next_sound_voiced	f0	f1	f2	f3	smiled
1	IY	true	200	300	450	500	true

Table 1. Predicted example record from database

The next step will be to divide this dataset into Training data and Testing data. I will aim for a ratio of 80:20 training to testing data, as is standard in supervised machine learning. I will then build and test a classifier by following guides from the Scikit-Learn resource page.

I will use PyCharm, Jupyter Notebooks and Markdown to develop a program that is interactive, where the user is asked to input measurements that will be used as parameters of the classifier function, the result will be a label (SS or NSS) for the sound described by the input measures.

As a summary, these are the phases I predict this project will comprise:

1. Data collection
2. Data cleaning
3. Database building
4. Classifier building
5. Classifier testing
6. Debugging and streamlining
7. Jupyter Notebook setup and documentation