Contrastive phonetic features ontology for automatic learning of allomorphs for unsupervised morphological parsing

Gina (Virginia) Cook
Engineering and Computer Science
Concordia University
Montréal, QC Canada
ginacook@alumni.concordia.ca

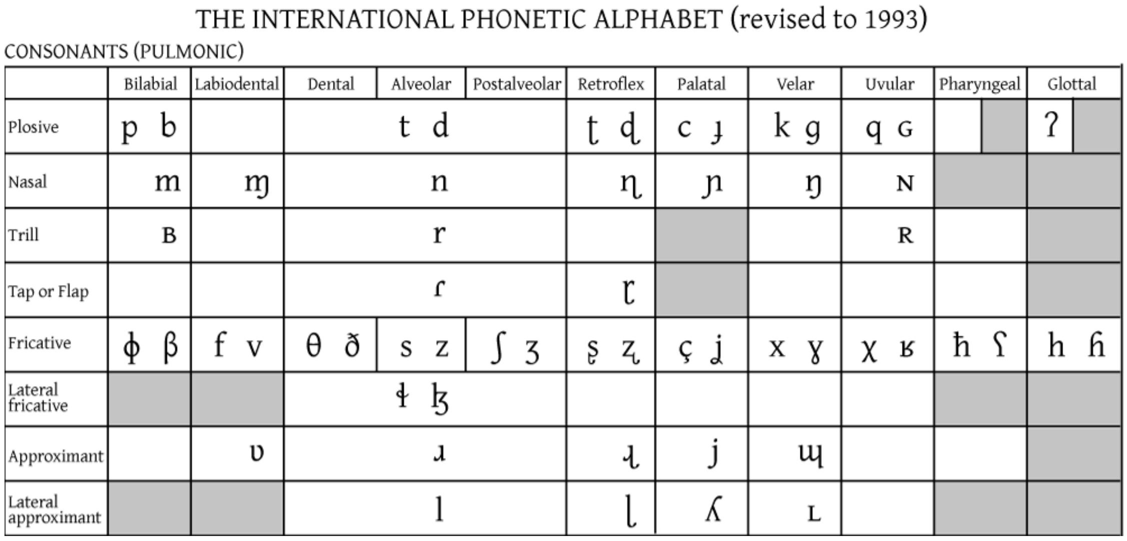
Abstract

Normally, languages (such as most Latinate and Germanic Languages) do not include predictable information in the orthography. However, some writing systems include predictable and systematic information which causes exponential increase of vocabulary size. This introduces problems of computation time and complexity when doing any Natural Language Processing task such as Information Retrieval. Fortunately this predictable information is well understood by phonologists who work on these languages. This paper discusses an ontology which encodes a the phonological, phonetic and orthographic concepts and relations which a Phonologist reasons over to discover the predictable systematicities in a languages spoken or written stream. More importantly this ontology can operate on any UTF-8 formatted text and return possible systematicities which can be investigated and removed from the vocabulary set to reduce the vocabulary size. The focus of the paper is on Inuktitut, one such language where written vocabulary raises exponentially due to predictable systematicities in the orthography.

Contents

1 Introduction

This paper serves to document the methods, results, and lessons learned in attempting to build a phonological ontology (Phonont.v1) which can be populated with any UTF-8 text data and provide some meaningful query services. The ontology's Abox was built and tested using UTF-8 data from a variety of languages of a variety of corpus size and type. The ontology's Tbox is composed of two primary knowledge sources, one prepared mostly by hand from


THE INTERNATIONAL PHONETIC ALPHABET (revised to 1993)

CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ	▨	ʔ ▨
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ	▨	▨
Trill	ʙ			r			▨		ʀ		▨
Tap or Flap				ɾ		ɽ	▨				▨
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative	▨	▨		ɬ ɮ						▨	▨
Approximant		ʋ		ɹ		ɻ	j	ɰ			▨
Lateral approximant	▨	▨		l		ɭ	ʎ	ʟ		▨	▨

Figure 1: International Phonetic Alphabet is organized using a Table which hides a more Ontologically complex network of concepts

literature on Phonological literature, the other automatically extracted from the corpora. Wherever possible information was automatically generated so as to replicate the process and discover and capitalize on systematicities in human reasoning when accumulating and integrating knowledge.

2 The Problem - Why model the ontology of Phonological concepts

The overarching goal behind Phonont.v1 is to explore both the possibilities and potential roadblocks in Ontological modeling (§ 2.1), its secondary goal is to understand ways of automating a phonological reasoner for tasks in Natural Language Processing which would benefit from removing phonological systematicities which cannot be detected based on letters alone (§ 4.1).

2.1 Lessons learned about the Semantic Web

The Semantic Web and Ontologies or Graph representations in general are very natural ways of encoding Taxonomic relations between concepts (such as *"/p/ isA Bilabial*), as well as more “meaningful” relations between concepts (such as /p/ isSimilarTo /b/).

2.1.1 Roles vs. Concepts

Phonont.v1 uses a great deal of Taxonomic relations as this is traditionally how phonological features are conceptualized as shown in the IPA table (fig. 1) where the “concepts” are found along the row and column labels and “individuals” are found in the intersection of these concepts. As can immediately be seen from this table, in fact the cells of the table may be represented as concepts, and the actual words of a language could be used as individuals. This can be yet further zoomed to consider the words of the language to be concepts, and the individual utterances as individuals.

The table in the IPA is an oversimplification of the concepts involved in human speech sounds. Phonont.v1 aims to unify concepts from various Phonological frameworks (SPE, Halle and Chomsky 1968, Feature Geometry Vaux 2000, ArticulatoryPhonetics, Keating and Lahiri 1989, Goldsmith 1990). A section of the inferred model of the hand built Phonology Tbox which is part of Phonont.v1 is shown in (fig. 2). This process of attempting to unify concepts across frameworks serves to highlight research questions which are being asked, and also provide the conceptual structure to make queries of documents to better understand these questions.

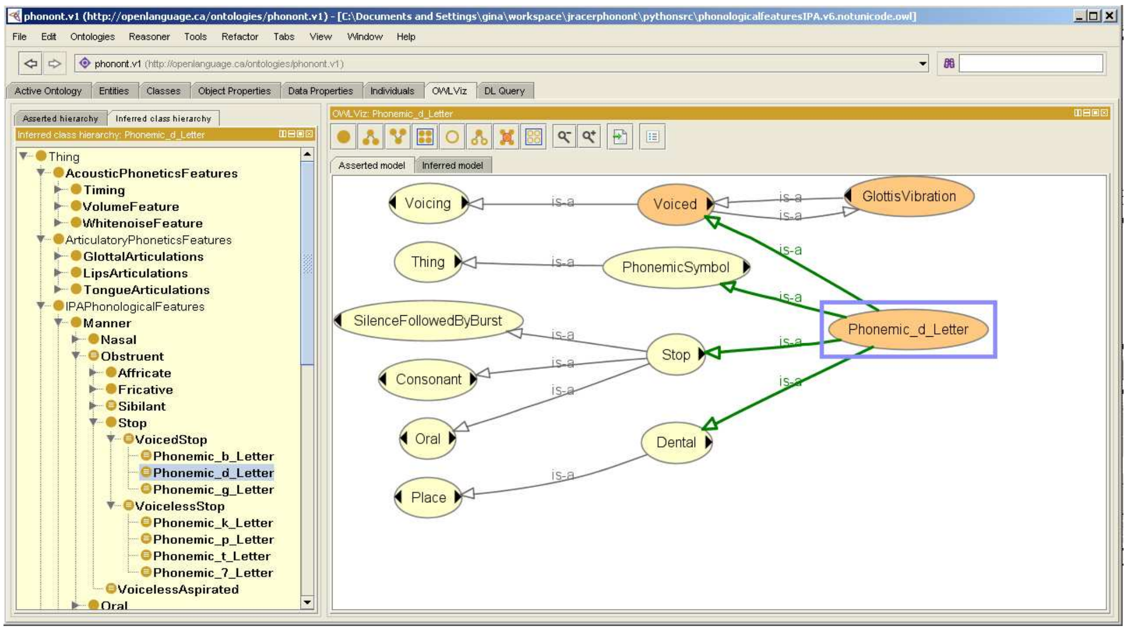

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\phonologicalfeaturesIPA.v6.notunicode.owl]
File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help
phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)
Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query
Asserted hierarchy Inferred class hierarchy
Inferred class hierarchy: Phonemic_d_Letter
Thing
AcousticPhoneticsFeatures
Timing
VolumeFeature
WhitenoiseFeature
ArticulatoryPhoneticsFeatures
GlottalArticulations
LipsArticulations
TongueArticulations
IPAPhonologicalFeatures
Manner
Nasal
Obstruent
Affricate
Fricative
Sibilant
Stop
VoicedStop
Phonemic_b_Letter
Phonemic_d_Letter
Phonemic_g_Letter
VoicelessStop
Phonemic_k_Letter
Phonemic_p_Letter
Phonemic_t_Letter
Phonemic_?_Letter
VoicelessAspirated
Oral
OWLViz: Phonemic_d_Letter
Asserted model Inferred model
Voicing
Voiced
GlottisVibration
Thing
PhonemicSymbol
SilenceFollowedByBurst
Phonemic_d_Letter
Stop
Consonant
Oral
Dental
Place
is-a

Figure 2: Phonological Ontology provides opportunities to unify and ask questions about the distinction between concepts across phonological frameworks

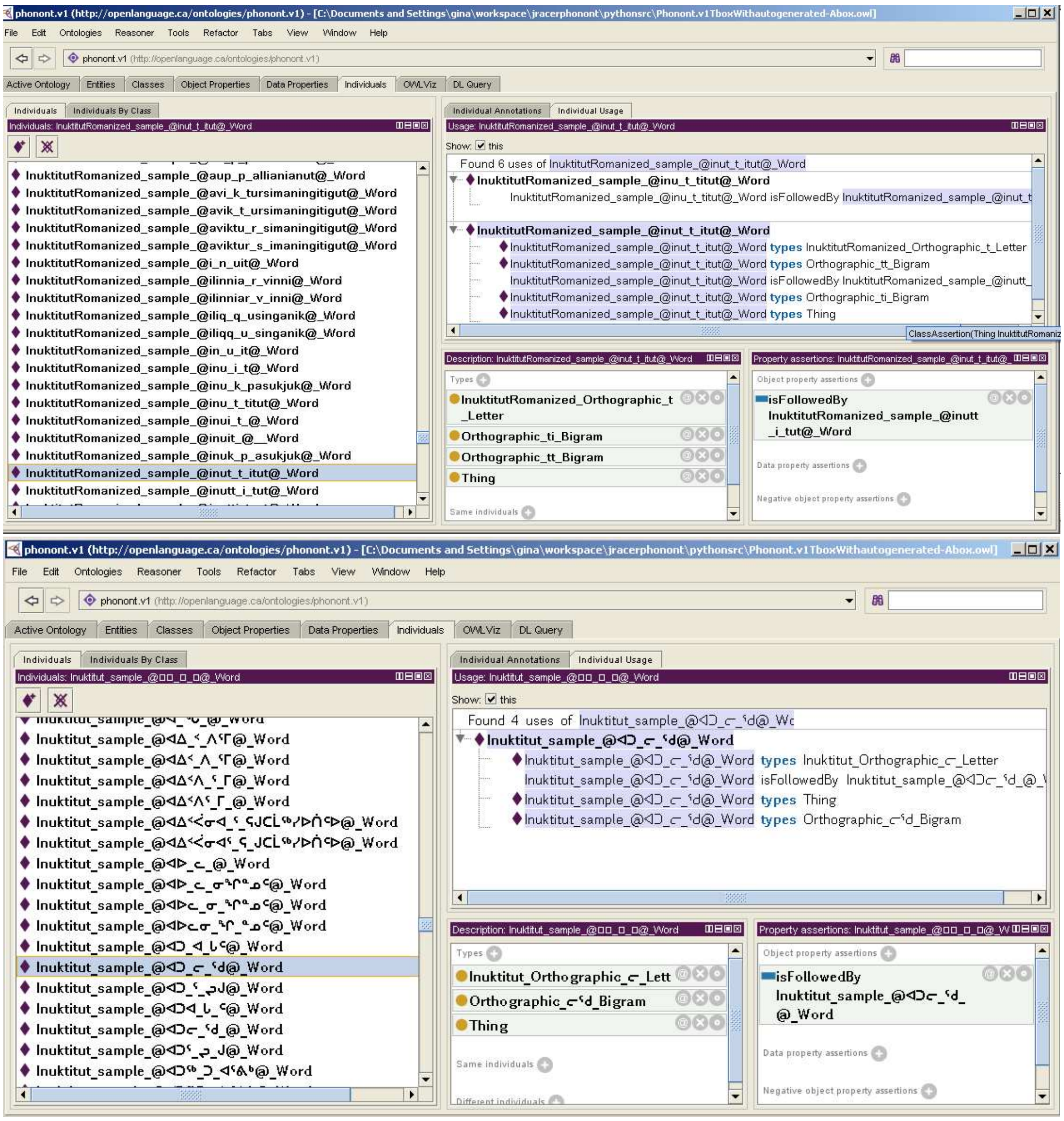
2.1.2 Nominals

How to model nominals (components which can be modeled as either individuals or concepts) a topic which can greatly increase the time complexity of a Semantic Web Reasoner. The RacerPro reasoner used in Phonont.v1 does not support nominals (or rather, approximates nominals). For both the reasons of complexity and support nominals were not used in Phonont.v1. Where to draw the line in the zoom is a tricky question. After some experimentation with different levels of zoom it was decided to use words in a corpus as individuals and letters as concepts.

The choice of encoding an idea as a concept, *LettersSimilarToP:* $\{b,m,t,k\}$ or as a role *p isSimilarTo b*, *p isSimilarTo m* ... is directly related to the types of individuals available in the system. I struggled to not use letters as individuals so that I could use roles to connect them. As shown in fig. 3 Individuals of a letter class were represented by a sample word from the corpus, in this case The Inuk Magazine corpus of Romanized Inuktitut (Inuktitut in Roman letters). While Phonont.v1 was designed to support Inuktitut Syllabics ultimately 1GB of RAM was not Enough to include the Syllabics in the Ontology and have the Pellet Reasoner classify (to visualize the progress in Inferred information during development. The Semantic Web does not allow roles to connect concepts. The only roles that can concept concepts are ‘taxonomic’ roles essentially subsumption, but can also be similar ideas in other domains such as *isA, isPartOf, isATypeOf, isAKindOf, impiles, isSubsumedBy*.

2.1.3 Ontologies and Lattices

In their discussion on using Ontologies to encode publications and research Tho et a. 2006 merge the ideas of Clustering and Fuzzy Logic in Text Classification with the TBox and Abox ideas of Semantic Web. Their formalisms for the Fuzzy Ontology Generation frAmework (FOGA) are very reminicent of joined semi-latices used in Set Theory Semantics (Link 1983). An approach which uses sets rather than truth values to calculate the meaning of sentences. In fact, Ontologies are much like latices, where sets are connected via the $\sqsupseteq$ operator, however not all sets are connected to other sets. When the individuals of a Ontology are viewed this way it can become clear why additional individuals cause exponential growth in classification time. The lattices in fig. 4 show the transitivity of the implication/entailment/subset relationship. The lattices also show why it is important to have an individual asserted in a concept if that concept must be used in reasoning.


phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\Phonont.v1TboxWithautogenerated-Abox.owl]

File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLViz | DL Query

Individuals | Individuals By Class

Individuals: InuktitutRomanized_sample_@inut_t_itut@_Word

```
InuktitutRomanized_sample_@aup_p_allianianut@_Word
InuktitutRomanized_sample_@avi_k_tursimaningitigut@_Word
InuktitutRomanized_sample_@avik_t_ursimaningitigut@_Word
InuktitutRomanized_sample_@aviktu_r_simaningitigut@_Word
InuktitutRomanized_sample_@aviktur_s_imaningitigut@_Word
InuktitutRomanized_sample_@i_n_uit@_Word
InuktitutRomanized_sample_@ilinnia_r_vinni@_Word
InuktitutRomanized_sample_@ilinniar_v_inni@_Word
InuktitutRomanized_sample_@iliq_q_usinganik@_Word
InuktitutRomanized_sample_@iliqq_u_singanik@_Word
InuktitutRomanized_sample_@in_u_it@_Word
InuktitutRomanized_sample_@inu_i_t@_Word
InuktitutRomanized_sample_@inu_k_pasukjuk@_Word
InuktitutRomanized_sample_@inu_t_titut@_Word
InuktitutRomanized_sample_@inui_t_@_Word
InuktitutRomanized_sample_@inuit_@__Word
InuktitutRomanized_sample_@inuk_p_asukjuk@_Word
InuktitutRomanized_sample_@inut_t_itut@_Word
InuktitutRomanized_sample_@inutt_i_tut@_Word
```

Individual Annotations | Individual Usage

Usage: InuktitutRomanized_sample_@inut_t_itut@_Word

Show: ☑ this

```
Found 6 uses of InuktitutRomanized_sample_@inut_t_itut@_Word
InuktitutRomanized_sample_@inu_t_titut@_Word
    InuktitutRomanized_sample_@inu_t_titut@_Word isFollowedBy InuktitutRomanized_sample_@inut_t
InuktitutRomanized_sample_@inut_t_itut@_Word
    InuktitutRomanized_sample_@inut_t_itut@_Word types InuktitutRomanized_Orthographic_t_Letter
    InuktitutRomanized_sample_@inut_t_itut@_Word types Orthographic_tt_Bigram
    InuktitutRomanized_sample_@inut_t_itut@_Word isFollowedBy InuktitutRomanized_sample_@inutt_
    InuktitutRomanized_sample_@inut_t_itut@_Word types Orthographic_ti_Bigram
    InuktitutRomanized_sample_@inut_t_itut@_Word types Thing
```

ClassAssertion(Thing InuktitutRomaniz

Description: InuktitutRomanized_sample_@inut_t_itut@_Word

Types
- InuktitutRomanized_Orthographic_t_Letter
- Orthographic_ti_Bigram
- Orthographic_tt_Bigram
- Thing

Same individuals

Property assertions: InuktitutRomanized_sample_@inut_t_itut@_

Object property assertions
- isFollowedBy InuktitutRomanized_sample_@inutt_i_tut@_Word

Data property assertions

Negative object property assertions

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\Phonont.v1TboxWithautogenerated-Abox.owl]

File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLViz | DL Query

Individuals | Individuals By Class

Individuals: Inuktitut_sample_@□□_□_□@_Word

```
Inuktitut_sample_@ᐊᐃ_ᑉ_ᐱᕐᒥ@_Word
Inuktitut_sample_@ᐊᐃᑉ_ᐱ_ᕐᒥ@_Word
Inuktitut_sample_@ᐊᐃᑉᐱ_ᕐ_ᒥ@_Word
Inuktitut_sample_@ᐊᐃᑉᐱᕐ_ᒥ_@_Word
Inuktitut_sample_@ᐊᐃᑉᐹᓂᐊ_ᕐ_ᕋᒍᑕᒫᖅᓯᐅᑏᕐᐅ@_Word
Inuktitut_sample_@ᐊᐃᑉᐹᓂᐊᕐ_ᕋ_ᒍᑕᒫᖅᓯᐅᑏᕐᐅ@_Word
Inuktitut_sample_@ᐊᐅ_ᓚ_@_Word
Inuktitut_sample_@ᐊᐅ_ᓚ_ᓂᖏᓐᓄᑦ@_Word
Inuktitut_sample_@ᐊᐅᓚ_ᓂ_ᖏᓐᓄᑦ@_Word
Inuktitut_sample_@ᐊᐅᓚᓂ_ᖏ_ᓐᓄᑦ@_Word
Inuktitut_sample_@ᐊᑐ_ᐊ_ᒐᑦ@_Word
Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Word
Inuktitut_sample_@ᐊᑐ_ᕐ_ᓗᒍ@_Word
Inuktitut_sample_@ᐊᑐᐊ_ᒐ_ᑦ@_Word
Inuktitut_sample_@ᐊᑐᓕ_ᑦᑯ_@_Word
Inuktitut_sample_@ᐊᑐᕐ_ᓗ_ᒍ@_Word
Inuktitut_sample_@ᐊᑐᖅ_ᑐ_ᐊᕐᕕᒃ@_Word
```

Individual Annotations | Individual Usage

Usage: Inuktitut_sample_@□□_□_□@_Word

Show: ☑ this

```
Found 4 uses of Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Wc
Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Word
    Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Word types Inuktitut_Orthographic_ᓕ_Letter
    Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Word isFollowedBy Inuktitut_sample_@ᐊᑐᓕ_ᑦᑯ_@_\
    Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Word types Thing
    Inuktitut_sample_@ᐊᑐ_ᓕ_ᑦᑯ@_Word types Orthographic_ᓕᑦᑯ_Bigram
```

Description: Inuktitut_sample_@□□_□_□@_Word

Types
- Inuktitut_Orthographic_ᓕ_Lett
- Orthographic_ᓕᑦᑯ_Bigram
- Thing

Same individuals

Different individuals

Property assertions: Inuktitut_sample_@□□_□_□@_W

Object property assertions
- isFollowedBy Inuktitut_sample_@ᐊᑐᓕ_ᑦᑯ_@_Word

Data property assertions

Negative object property assertions

Figure 3: Example of individuals and Relations in Phonont.v1

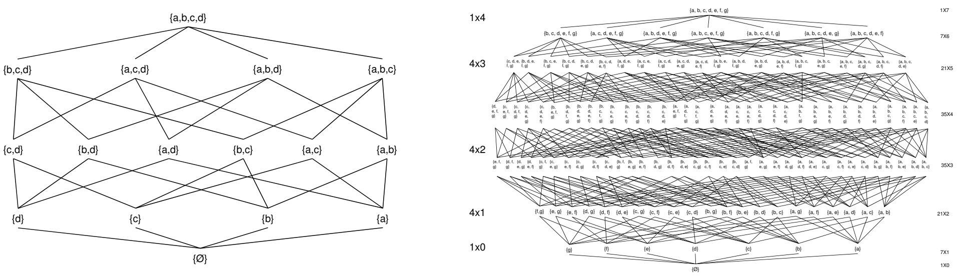


Figure 4: Truth can be viewed as the presence or absence of members in the set. Compare the complexity of fully connected lattices with 4 individuals and a lattice with 7.

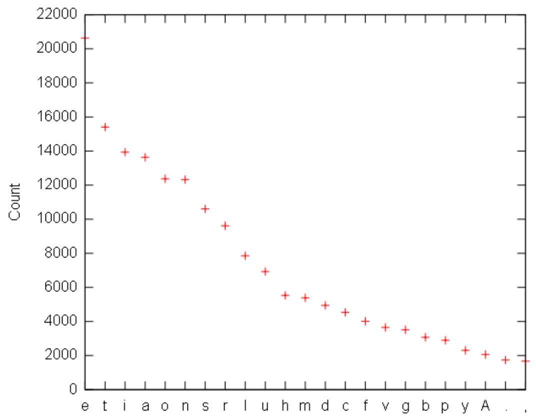


Figure 5: Sample graph of ZIpf's law for letter frequencies, used to create Fuzzy sets of VeryCore, Core and NotCor letters per language.

2.1.4 Ontologies and Uncertainty

One of the most important problems in building a Phonological reasoner is reasoning with uncertain edges of concepts. This has been identified as a research area in Semantic Web. Much of Human reasoning includes likleyhoods of truth, or degrees of membership such as very expensive, hot beach (what does hot mean?) (Stoilos et al. 2005). There are a few approaches to uncertainty, including Baisian Nets and Fuzzy Logic.

One of the most important problems in building a Phonological reasoner is reasoning with uncertain edges of concepts. This has been identified One of the tasks in building Phonont.v1 was to Populate the ontology with letters of the corpus. This feature of Phonont.v1 can later be used in Language Recognition to determine the Language source of words in a document which is multi-lingual (a rising problem in most languages where English words are sprinkled among native words). A fuzzy logic implementation of "coreness" was used to encode this valuable information. After running the Python script which extracts letter counts from texts, letter frequencies were graphed and visualized using GnuPlot (fig. **??**). Letters from other languages (such as French letters appearing in English or Inuktitut texts) appeared in the bottom tail of the letter frequencies. Letters which appeared less than 1% of the most frequent letter were labled as NonCore letters for the language of that Corpus. Letters which appeared in the top 30% were labeled as VeryCore. VeryCore letters could serve in langauge recognition but also later in in determining which phonotactic clusters should be present but are missing from the data.

2.1.5 The Semantic Web and Internationalization (Unicode Struggles)

The vast majority of man hours behind the Phonont.v1 development went into dealing with quirks of text Encoding. Both in the software, XML and the corpora sources themselves (the Inuktitut Corpus was generated by spidering two different websites then transcoding the information). RacerPro 2.0 now supports UTF-8 unicode encoding. Jracer, implemented in Java also supports UTF-8. Protege 4 also supports UTF-8 encoding. Python also supports UTF-8 but will complain profusly in Windows environment and less in a Linux environment. Yet mysteriously off and on data would become unreadable, possibly due to developing between multiple machines, operating systems, text editors and command prompt/IDE environments. Ultimately most of the development was done in Eclipse on Windows as getting RacerPorter to run, and JRacer to run was rather straightforward. After prolific testing of JRacer with various UTF-8 texts to successful results, development went ahead using Eclipse, Java and Protege to build the Phonont Tboxes and Aboxes. While the end result is a Ontology which can be autogenerated and visualized, it will not load in RacerPro anymore. And if it is saved from Protege the UTF-8 encoding is overwritten with XML escape sequences and so will no longer display in the RacerPro output.

After much development it was discovered that source of continuous errors of perfectly valid UTF-8 and OWL were due to XML. According to errors in Protege, XML does not seem to allow UTF-8 in comments, more specifically in the URI which expresses the location of resources in OWL. By the end of development the pruning of punctuation and UTF-8 characters which the XML objected to would take over 20 minutes to test while Phonont was being generated. In a future implementation I will explore writing the Ontology in Racer's language rather than XML and

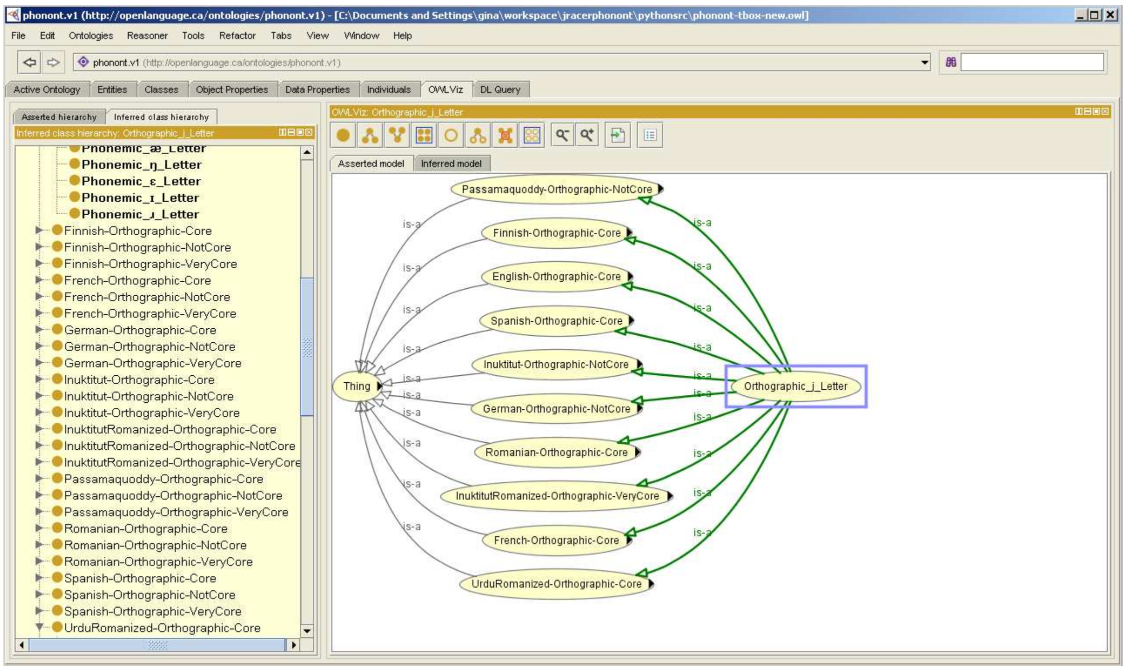

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\phonont-tbox-new.owl]
File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help
phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)
Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query
Asserted hierarchy Inferred class hierarchy
Inferred class hierarchy: Orthographic_j_Letter
Phonemic_æ_Letter
Phonemic_ŋ_Letter
Phonemic_ɛ_Letter
Phonemic_ɪ_Letter
Phonemic_ɹ_Letter
Finnish-Orthographic-Core
Finnish-Orthographic-NotCore
Finnish-Orthographic-VeryCore
French-Orthographic-Core
French-Orthographic-NotCore
French-Orthographic-VeryCore
German-Orthographic-Core
German-Orthographic-NotCore
German-Orthographic-VeryCore
Inuktitut-Orthographic-Core
Inuktitut-Orthographic-NotCore
Inuktitut-Orthographic-VeryCore
InuktitutRomanized-Orthographic-Core
InuktitutRomanized-Orthographic-NotCore
InuktitutRomanized-Orthographic-VeryCore
Passamaquoddy-Orthographic-Core
Passamaquoddy-Orthographic-NotCore
Passamaquoddy-Orthographic-VeryCore
Romanian-Orthographic-Core
Romanian-Orthographic-NotCore
Romanian-Orthographic-VeryCore
Spanish-Orthographic-Core
Spanish-Orthographic-NotCore
Spanish-Orthographic-VeryCore
UrduRomanized-Orthographic-Core
OWLViz: Orthographic_j_Letter
Asserted model Inferred model
Passamaquoddy-Orthographic-NotCore
Finnish-Orthographic-Core
English-Orthographic-Core
Spanish-Orthographic-Core
Inuktitut-Orthographic-NotCore
Thing
Orthographic_j_Letter
German-Orthographic-NotCore
Romanian-Orthographic-Core
InuktitutRomanized-Orthographic-VeryCore
French-Orthographic-Core
UrduRomanized-Orthographic-Core
is-a

Figure 6: Fuzzy Set Membership can serve to encode relative "importance" of automatically generated data.

create a separate OWL XML generator which operates on this. As the code behind Phonont is modular this requires only changing one line in the population script, and perhaps adding a housekeeping array to declare all entitites in the ontology.

3 Generating the Ontology

The ontology can be generating using the Java Class Text2Ontology.java included in the source of the ontology at http://openlanguage.ca/ontologies/phonont.v1/ The Java class begins by calling several Python scripts which extract and count words in the corpora, extract and count letters, and bigrams, and finally populate mini-Tbox/aboxes using the letters, bigrams and words for each corpus. These Tboxes can then be later selected for addtion into the base TBox, along with the appropriate phoneme mapping for that language (also automatically generated by TBoxConstruction.java).

3.1 Knowledge Extraction

When the entire corpus is run and populated there are over 15000 assertions and it takes over 20 minutes to generate. Of course, normally only the Tboxes are needed in the final Phonont, the Aboxes can be added when the Ontology is being queried about a particular set of languages. Below is the output of Text2Ontology.java to demonstrate the control flow of the Ontology generation process. In addition to the corpora included in the base Tbox below there is also an Abox and Tbox for Korean, however given its syllabic orthography generates over 1000 classes in just the Tbox alone, compared to most other languages which range between 20-40 concepts for their Tbox (most of which overlap with the other European languages).

```
Language is English
File is br-phono.txt-unicode.txt
('    Total words in file: ', 33399)
('    Most Frequent letter ,', 353)
('    Most Frequent bigram ,', 904)
Language is German
```

```
File is german-mommsen-Roemische_Geschichte-Book01.utf8.txt
('    Total words in file: ', 86664)
('    Most Frequent letter ,', 12995)
('    Most Frequent bigram ,', 52309)
Language is English
File is english-1_Harry_Potter_and_the_Sorcerers_Stone.txt
('    Total words in file: ', 77601)
('    Most Frequent letter ,', 4048)
('    Most Frequent bigram ,', 8466)
Language is French
File is french-1_Harry_Potter_et_la_Pierre_Philosophale.utf8.v2.txt
('    Total words in file: ', 85889)
('    Most Frequent letter ,', 6290)
('    Most Frequent bigram ,', 13900)
Language is Spanish
File is spanish-1_Harry_Potter_y_la_Piedra_Filosofal-cleaned.utf8.txt
('    Total words in file: ', 78177)
('    Most Frequent letter ,', 5915)
('    Most Frequent bigram ,', 13231)
Language is Passamaquoddy
File is passamaquoddy-TalesFromMaliseet.utf8.txt
('    Total words in file: ', 17393)
('    Most Frequent letter ,', 2787)
('    Most Frequent bigram ,', 12060)
Language is UrduRomanized
File is urdu-forum-060427aagar.aap.ke.pechey.kotta.lag.jaey.v4-notimes.txt
('    Total words in file: ', 5987)
('    Most Frequent letter ,', 1155)
('    Most Frequent bigram ,', 1493)
Language is Romanian
File is romanian-ziare.0.v5sentences-cleaned.utf8.txt
('    Total words in file: ', 339506)
('    Most Frequent letter ,', 23506)
('    Most Frequent bigram ,', 51121)
Language is InuktitutRomanized
File is InukMagazine102-104rough-inuktitut.utf8.txt
('    Total words in file: ', 17874)
('    Most Frequent letter ,', 9293)
('    Most Frequent bigram ,', 42402)
Language is Inuktitut
File is assembly.nu.news-innuktitut.txt
('    Total words in file: ', 9764)
('    Most Frequent letter ,', 2890)
('    Most Frequent bigram ,', 40762)
Language is Finnish
File is leipzig-fi.small.txt
('    Total words in file: ', 406791)
('    Most Frequent letter ,', 82688)
('    Most Frequent bigram ,', 307525)
('    The number of assertions written to file: ', 15512)
```

4 Competency Questions

4.1 What can a Phonologist do?

4.1.1 Allophonic Rules

English speakers usually think they are saying a ’s’ (a *phoneme* is defined as a mental representation of a speech sound in one’s language) in the words /statement**s**/ and /algorithm**s**/, when they might be pronouncing a ’s’ [statement-s] or ’z’[algorithm-z] (*allophones* are defined as the variant realizations of a phoneme when produced) depending on the sounds around it. Given that speech segments are created in physical space the muscles involved are not independent and as a result a speech sound will be produced differently depending on the sounds around it.

4.1.2 Contrastive Features

In addition, if a certain muscle is not used in a language to make a contrastive difference (two words have different meanings) then this muscle is not specified for all speech sounds and can serve to augment the distinctions already available. For example vibration of the vocal cords are not used to distinguish sounds in Korean or Inuktitut.

In addition whether or not a muscle (or phonological feature) is contrastive can depend on the features’s position in an utterance. In German voicing is not contrastive at the end of a word, in English voicing is not contrastive at the end of a phonogical phrase. Phoneticians/phonogists gain experience in contrastive features and feature changing processes in different positions of a word. These changes are important in-order to uncover the underlying morpheme and reduce vocabulary size of morphemes.

5 Conclusion

In future work I will re-write the ontology generation files to allow for garenteed plugin to Racer. As the ontology stands now it must loose UTF-8 fonts in order to be processed and reordered by Protege so that Racer can read it. In the course of this project I gained a surprising amount of experience with XML, regular expressions, Python, Java and Racer. However, the amount of technical experience and the debugging process really forced me to think about what was underlying my project, how can I capture the semi-statistical ideas behind natural human reasoning of a “prototypical” letter of Inuktitut, vs Spanish? How can I capture the phonotactics of Spanish as compared to English (which allows lots of consonant clusters)? These are questions that Phonont.v1 answers visually, with a few mouse clicks. It’s exciting to imagine the possibilities of running a powerful Reasoner on it to extract more pairs and to investigate the original questions which motivated the construction of Phonont.

References

[1] Antoniou, Grigoris and Van Harmelen, Frank. 2008. *A semantic web primer.* Second edition. Cambridge, MA: The MIT Press.

[2] Calvanese, D. et al. 2007. “Software Tools for Ontology Access, Processing, and Usage.” *Deliverable TONES-D21.* Thinking ONtologiES. www.tonesproject.org

[3] Kamholz, David. 2006. “An Ontology for Sounds and Sound Patterns.” Max Planck Institute For Evolutionary Anthropology, Leipzig. http://emeld.org/workshop/2005/papers/kamholz-paper.html

[4] Konstantinou, N. et al. 2008. “Ontology and database mapping: a survey of current implementations and future directions.” *Journal of Web Engineering,* Vol. 7, No.1 (001-024).

[5] Link, Godehard. 1983. “The Logical Analysis of Plurals and Mass Terms : A Lattice-theoretical Approach.” In *Meaning, Use and Interpretation of Language,* R. Bauerle, C. Schwarze and A. von Stechow eds. Gruyter, New York.

[6] Protege Wiki http://protegewiki.stanford.edu/

[7] RacerPro Reference Manual 1.9.2 BETA http://www.racer-systems.com/

[8] RacerPro Users Guide 1.9.2 BETA http://www.racer-systems.com/

[9] Sene, S. and A. Shirke. 2009. “Generating OWL ontologies from a relational databases for semantic web.” *International Conference on Advances in Computing, Communication and Control (ICA C3’09)*

[10] Stoilos, Giorgos et al. 2005. “Fuzzy OWL: Uncertainty and the Semantic Web.” *Proc. Intl Workshop OWL: Experiences and Directions.* http://image.ntua.gr/papers/398.pdf

[11] Tho, Quan Thanh, et al. 2006. “Automatic Fuzzy Ontology Generation.” IEEE Transactions in Knowledge and Data Engineering, Vol.18, No.6.