Supervised Methods for Automatic Acronym Expansion in Medical Text

Mahesh Joshi
Department of Computer Science, University of Minnesota Duluth
Summer 2005 Intern, Division of Biomedical Informatics, Mayo Clinic

25th August 2005

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Terminology

- **Abbreviation**[1]: *a shortened form of a written word or phrase used in place of the whole*
  - e.g. *AcG* for *accelerator globulin*
- **Acronym**[2]: *a word formed from the initial letter or letters of each of the successive parts or major parts of a compound term*
  - e.g. *CC* for *common cold*
- Every acronym is an abbreviation, not vice-versa

1,2: Definitions from the Merriam Webster Online Dictionary (http://www.m-w.com/)

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

The Problem

- Acronyms and Abbreviations are widely used in clinical notes
- Their widespread use for various terms gives rise to ambiguity among them
  - e.g. *AC* can mean:
    - ***A****ntitussive with* ***C****odeine* – a cough medicine and/or a pain reliever
    - ***A****cromio****c****lavicular* – relating to the joint formed between the acromion and the clavicle
    - ***A****cid* ***C****ontroller* – a drug used to treat peptic ulcers and gastritis and esophageal reflux
    - any of the 13 different senses we have encountered

Information Retrieval

- Ambiguity among acronyms can be a significant problem in medical information retrieval (IR)
- In IR, augmenting search query with acronyms of search terms can enhance performance
- Consider the following numbers obtained from 17,056,336 notes representing 993,721 patients
  - e.g. ACA –
    - ACA only – 5,483 notes (2,543 patients)
    - 'adeno carcinoma' or 'adenocarcinoma' only – 299,714 notes (66,057 patients)
    - ACA and ('adeno carcinoma' or 'adenocarcinoma') – 1,209 notes (880 patients)

Information Retrieval

- e.g. DJD
  - DJD only – 175,956 notes (61,430 patients)
  - 'degenerative joint disease' only – 225,859 notes (78,428 patients)
  - DJD and 'degenerative joint disease' – 19,349 notes (12,856 patients)
- Augmenting the search with acronyms add ~2% (5483/ 299714) and ~77% (175956 / 225859) more documents to original search results for ACA and DJD, increasing the sensitivity or recall for the search.

The Problem

- Ambiguity of acronyms can degrade this performance by bringing down the specificity or precision of the search.
- ACA for example has 7 possible senses and the extra 5483 notes could contain the term ACA with any of those senses.
- Methods for automatic acronym expansion can therefore be employed for intelligent indexing of documents containing acronyms.

A Solution

- Treat automatic acronym expansion similar to word sense disambiguation (WSD)
- Use the surrounding context of the acronym to decide the correct sense, just like a human would
  - "The Robitussin **AC** doesn't affect his cough much …" - *antitussive with codeine*
  - "History of left supraspinatus tear and DJD of the left **AC** joint" – *acromioclavicular*
  - "Pepcid **AC** two every day" – *acid controller*

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Supervised Learning Methods

- The “state of the art” and a very popular approach to WSD, yielding high accuracy on this task
- Initially require a set of manually classified or “sense tagged” examples – known as the *training data*
- Using some *learning algorithm* and *features* from the training data, these methods generate a *classifier*
- The classifier can be used to classify future instances of test data

What do the algorithms learn

	Robitussin	cough	supraspinatus	joint	Pepcid	Sense
AC – 1	1	1	0	0	0	A
AC – 2	0	0	1	1	0	B
AC – 3	0	0	0	0	1	C
AC – 4	1	0	0	0	0	A
AC – 5	0	0	0	1	0	B

What do the algorithms learn

	Robitussin	cough	supraspinatus	joint	Pepcid	Sense
AC – 1	1	1	0	0	0	A
AC – 2	0	0	1	1	0	B
AC – 3	0	0	0	0	1	C
AC – 4	1	0	0	0	0	A
AC – 5	0	0	0	1	0	B

What do the algorithms learn

	Robitussin	cough	supraspinatus	joint	Pepcid	Sense
AC – 1	1	1	0	0	0	A
AC – 2	0	0	1	1	0	B
AC – 3	0	0	0	0	1	C
AC – 4	1	0	0	0	0	A
AC – 5	0	0	0	1	0	B

What do the algorithms learn

	Robitussin	cough	supraspinatus	joint	Pepcid	Sense
AC – 1	1	1	0	0	0	A
AC – 2	0	0	1	1	0	B
AC – 3	0	0	0	0	1	C
AC – 4	1	0	0	0	0	A
AC – 5	0	0	0	1	0	B

Choice of algorithms

- Support Vector Machines
  - Introduced by Vapnik (1995)
  - Discriminative method based on Perceptron learning
- The naïve Bayes classifier
  - Based on the Bayes' rule for conditional probabilities
  - Simplifying assumption of conditionally independent features
- Decision trees
  - Divide and conquer strategy, forming a tree of questions with "yes – no" answers, based on the available features
  - Crucial features near the root, selected using information gain measures

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Related Work

- Liu et al. (JAMIA 2004)
  - Fully supervised approaches using naïve Bayes classifier, decision lists, and their adaptation of decision list classifier
- Pakhomov (ACL 2002), Pakhomov et al. (AMIA 2005)
  - Unsupervised training data generation from Mayo clinical notes, MEDLINE collection and WWW + supervised disambiguation of abbreviations

Related Work

- Mohammad and Pedersen (CoNLL 2004)
  - Employ unigram, bigram and syntactic features
- Pedersen (NAACL 2000)
  - Uses ensembles of multiple naïve Bayes classifiers trained on unigrams in various window sizes

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Training Data Generation

- The biggest hurdle in supervised approaches – lack of sufficient hand labeled training data
- In our case, the focus was on analyzing machine learning algorithms with respect to several types of features
- Still, selecting the right kind of data for the annotation process done by the medical data retrieval experts is crucial

Important Considerations

- Choosing acronyms
  - Practical importance
  - Frequency
  - Sense distribution
- Sense Inventory – a list of possible expansions for the selected acronyms
  - UMLS listed expansions in LRABR table
  - Mayo Clinic approved expansions
  - Diagnosis codes from master-sheet data
    - Master-sheet entries are diagnostic statements about patients, and each master sheet entry is manually assigned an 8 digit diagnosis code from the Hospital Adaptation of the ICDA code (HICDA)

Acronym Finding

- Initially identified a set of 25 acronyms using UMLS sense inventory as reference
  - These had a highly skewed distribution in Mayo data
- Used the Mayo master-sheet data (22,705,083 diagnosis statements), with the following criteria to select an acronym:
  - Has two or more diagnosis codes associated with it in master-sheet, a diagnosis code is considered unique only if it differs in the first five digits out of eight from others
  - Has a relatively balanced distribution of the number of different diagnosis codes associated with it
  - Considered practically important by medical data retrieval experts
- Identified 7 acronyms which are being annotated

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Feature Engineering

- Different types of features used for WSD:
  - Bag of Words in context
  - Parts of Speech of words in context
  - Syntactic relationships (noun phrase, verb phrase, subject-object)
  - Collocations in context
  - Symbolic knowledge from an ontology such as UMLS or WordNet
  - Discourse level features such as section identifiers in clinical notes, e.g. CC *(Chief Complaint),* HPI *(History of Present Illness)*

Our features

- Unigrams in a flexible window of 1 to 10 around the acronym
- Two word collocations, i.e. bigrams in a flexible window of 1 to 10
- Parts of Speech of two words to the left and right of the acronym
- Clinical note features:
  - Service Code – *represents the department where the patient was treated (Cardiology, Rheumatology etc.)*
  - Gender Code
  - Section Id

Why medical features might help

- **APC**: *Atrial Premature Contraction (Cardiology)*, *Argon Plasma Coagulation (Gastroenterology)*
  - Service Code might help
- **AP**: *Angina Pectoris* is more commonly *diagnosed* among male population
  - Gender Code might help

Feature Identification Tools

- Annotated XML file generation from clinical notes: UIMA (Unstructured Information Management Architecture), http://www.research.ibm.com/UIMA/
- Tokenization, Part of Speech Tagging: ANNIE system (A Nearly-New Information Extraction system) in GATE (General Architecture for Text Engineering), http://gate.ac.uk
- Unigram and bigram features identification using frequency cutoff and log likelihood measure: NSP (Ngram Statistics Package), http://ngram.sourceforge.net/
- Machine Learning Algorithms Implementation: WEKA (Waikato Environment for Knowledge Analysis), http://www.cs.waikato.ac.nz/ml/weka/

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Results – Unigrams + Bigrams

	Majority	C 5.0	Maximum Entropy	Naïve Bayes	SVM	C 4.5
AC	31.40	94.60	**96.70**	96.34	95.91	95.47
ACA	87.40	93.10	97.00	**97.97**	97.78	95.75
APC	42.30	90.70	**95.90**	92.82	93.09	89.89
CF	76.30	95.80	94.20	96.90	**97.18**	95.63
HA	92.30	94.70	95.80	**97.45**	96.27	94.70
LA	88.50	92.60	94.60	**97.13**	96.93	94.06
NSR	99.00	98.80	99.00	**99.26**	99.01	99.01
PE	48.30	90.80	**93.30**	90.56	91.91	90.94

Results – U + B + POS

	Majority	C 5.0	Maximum Entropy	Naïve Bayes	SVM	C 4.5
AC	31.40	94.60	**96.70**	95.26 96.34	96.12 95.91	94.40 95.47
ACA	87.40	93.10	97.00	**97.97** **97.97**	**97.97** 97.78	95.01 95.75
APC	42.30	90.70	**95.90**	93.09 92.82	93.09 93.09	90.43 89.89
CF	76.30	95.80	94.20	97.04 96.90	**97.32** 97.18	95.21 95.63
HA	92.30	94.70	95.80	**97.84** 97.45	96.07 96.27	94.70 94.70
LA	88.50	92.60	94.60	97.13 97.13	**97.75** 96.93	95.90 94.06
NSR	99.00	98.80	99.00	98.27 **99.26**	99.01 99.01	99.01 99.01
PE	48.30	90.80	**93.30**	92.29 90.56	92.87 91.91	91.52 90.94

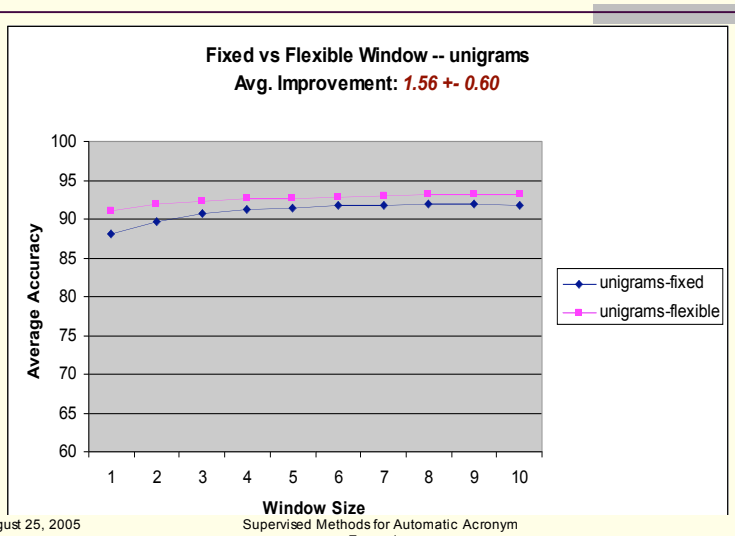
Results – U + B + CF

	Majority	C 5.0	Maximum Entropy	Naïve Bayes	SVM	C 4.5
AC	31.40	94.60	**96.70**	95.47 96.34	95.91 95.91	94.40 95.47
ACA	87.40	93.10	97.00	**98.15** 97.97	**98.15** 97.78	94.09 95.75
APC	42.30	90.70	**95.90**	93.09 92.82	93.35 93.09	90.43 89.89
CF	76.30	95.80	94.20	**97.46** 96.90	96.76 97.18	94.93 95.63
HA	92.30	94.70	95.80	**97.84** 97.45	97.45 96.27	94.89 94.70
LA	88.50	92.60	94.60	95.90 **97.13**	95.70 96.93	94.06 94.06
NSR	99.00	98.80	99.00	96.05 **99.26**	99.01 99.01	99.01 99.01
PE	48.30	90.80	93.30	92.29 90.56	**93.45** 91.91	91.52 90.94

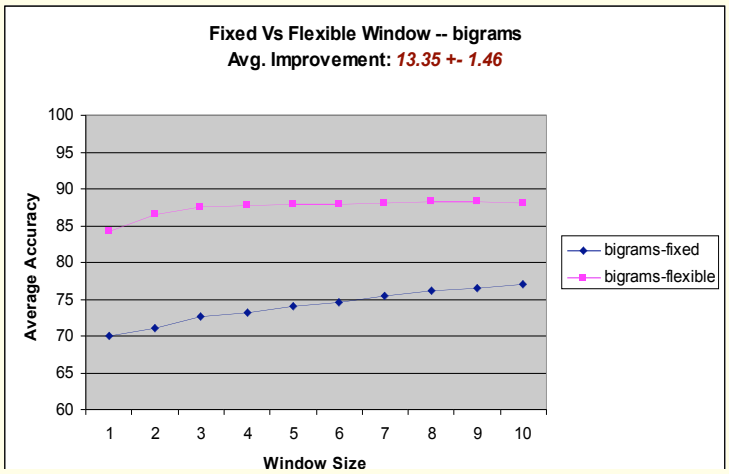
Results – U + B + POS + CF

	Majority	C 5.0	Maximum Entropy	Naïve Bayes	SVM	C 4.5
AC	31.40	94.60	**96.70**	95.26 96.34	96.34 95.91	94.40 95.47
ACA	87.40	93.10	97.00	97.97 97.97	**98.15** 97.78	94.09 95.75
APC	42.30	90.70	**95.90**	93.35 92.82	93.62 93.09	90.43 89.89
CF	76.30	95.80	94.20	97.32 96.90	**97.46** 97.18	94.93 95.63
HA	92.30	94.70	95.80	**97.84** 97.45	97.64 96.27	94.89 94.70
LA	88.50	92.60	94.60	97.13 97.13	**97.54** 96.93	95.90 94.06
NSR	99.00	98.80	99.00	97.53 **99.26**	**99.26** 99.01	99.01 99.01
PE	48.30	90.80	93.30	93.06 90.56	**93.45** 91.91	91.52 90.94

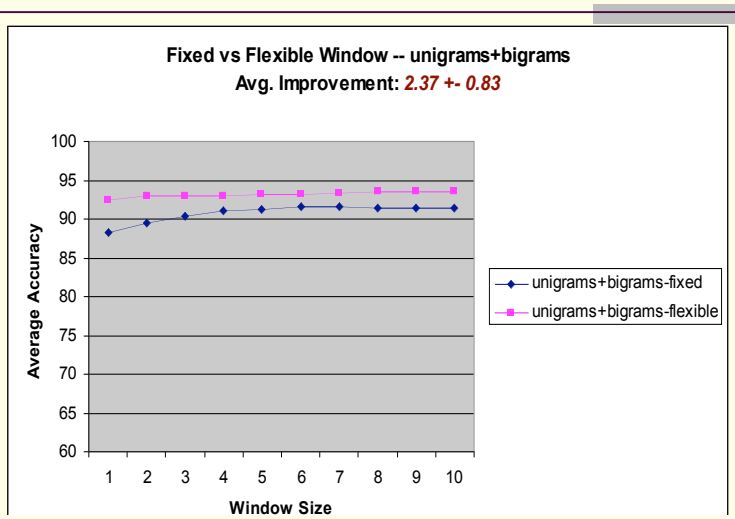
Fixed vs Flexible Window



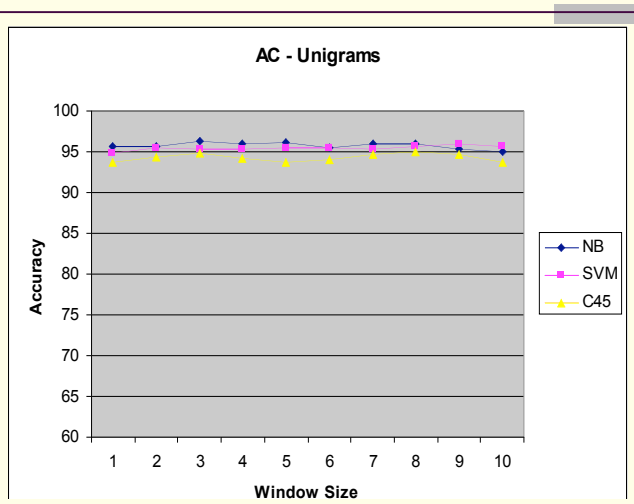
Fixed vs Flexible Window



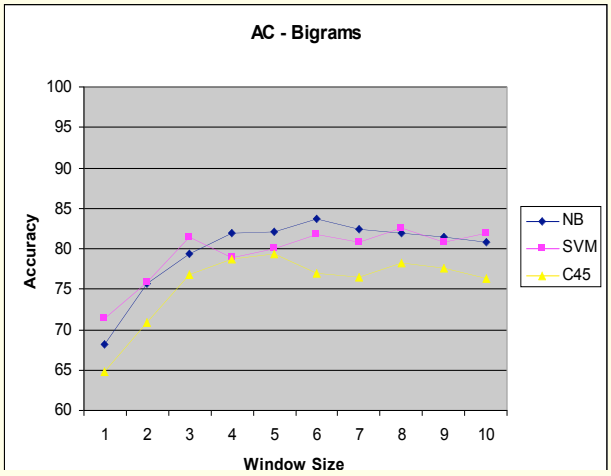
Fixed vs Flexible Window



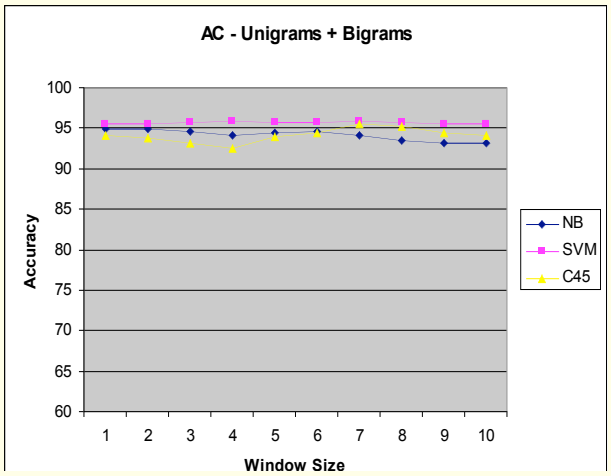
Learning Curve



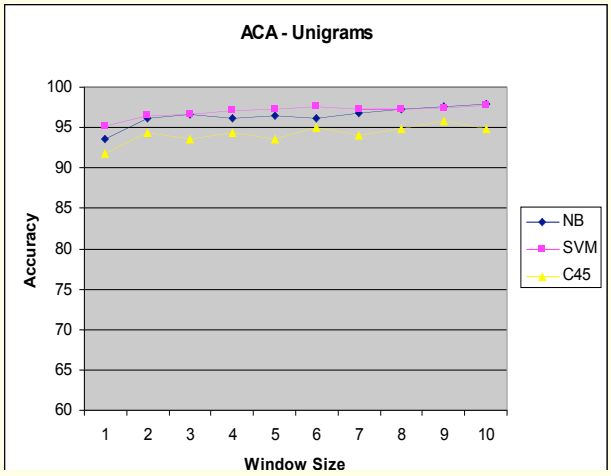
Learning Curve



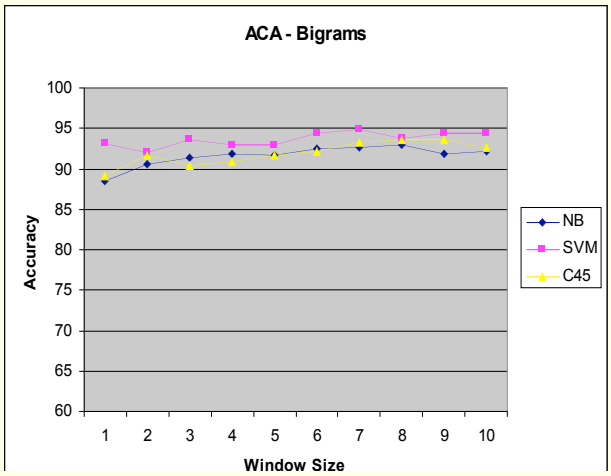
Learning Curve



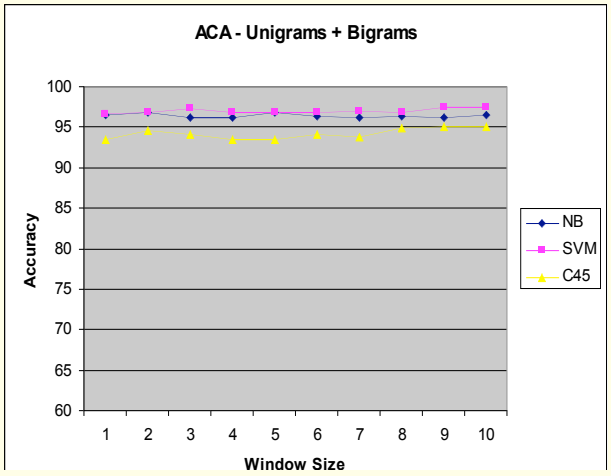
Learning Curve



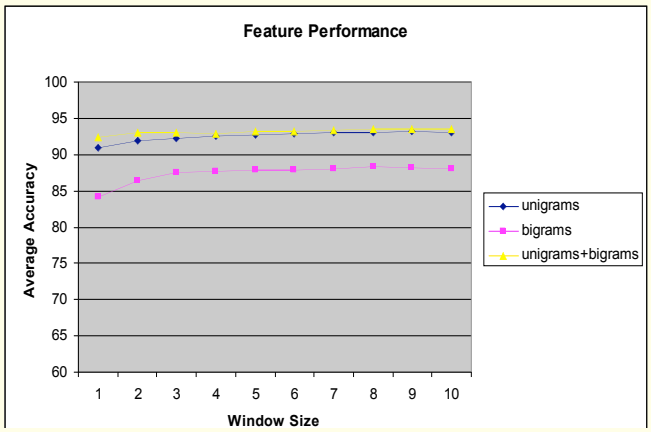
Learning Curve



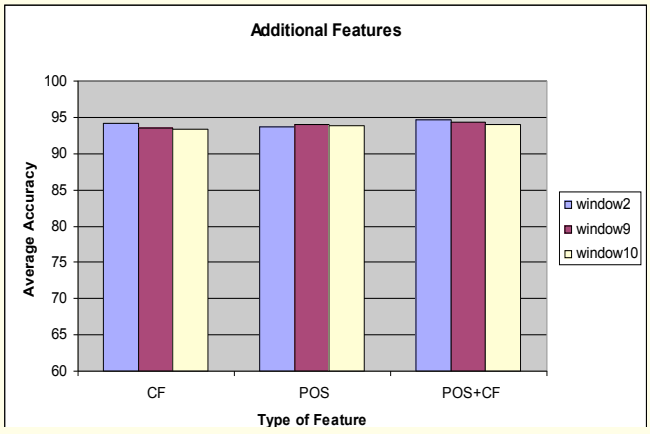
Learning Curve



Feature Performance



Additional Features



Overall Classifier Performance

	Accuracy (%)	Training Time (s)	Testing Time (s)
Naïve Bayes	91.57 ± 5.97	**0.66 ± 0.40**	7.62 ± 4.85
SVM	**93.26 ± 4.85**	1.48 ± 0.94	0.15 ± 0.11
C 4.5	90.33 ± 6.92	8.40 ± 6.12	**0.02 ± 0.01**

Findings

- Window size beyond 3 significant unigrams / bigrams does not seem to improve performance substantially
- SVMs were able to make better use of complimentary features
- Overall, two significant unigrams and bigrams on each side, POS and clinical features performed well for all classifiers

Outcomes

- Development of an annotation infrastructure that we can pursue further for other acronyms / ambiguous terms
- Framework for experimentation and testing of various supervised algorithms for WSD
- Uncovering the extent of the problem with acronym data generation from medical records
- The developed classifier models can be plugged into a UIMA-Weka interface

Overview

- Background
  - The Problem
  - Supervised Learning Methods
  - Related Work
- Methods
  - Training Data Generation
  - Feature Engineering
- Results
- Summary

Summary

- Acronym disambiguation is an important aspect in automatic text analysis
- Manually labeled training data generation for supervised methods is a complex task
  - Semi-supervised methods are attractive from this perspective
- Conventional WSD features perform quite well with acronym disambiguation, as expected
- Domain specific features like service code, gender code and section id improve results to some extent

Acknowledgements

- **Dr. Seguei Pakhomov** for his continual support and advice and for giving me the right level of independence in choosing the direction of work.
- **Dr. Ted Pedersen** and **Dr. Richard Maclin** from University of Minnesota, Duluth for their encouragement to pursue this internship and invaluable guidance in research.
- Medical data retrieval experts **Barbara Abbot**, **Debra Albrecht** and **Pauline Funk**, without whom this study would not have been possible at all!
- **Patrick Duffy** for his technical advice in various matters.
- **Dr. Guergana Savova** and **James Buntrock** for their feedback and questions that raised interesting issues.
- **Dr. Christopher G. Chute**

References

- **Commission on Professional and Hospital Activities – Hospital Adaptation of ICDA.** *2nd ed. Vol. 1. 1973, Ann Arbor, MI: Commission on Professional and Hospital Activities*
- **Liu H., Teller V. and Friedman C. – A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation**. *Journal of the American Medical Informatics Association (2004)*
- **Mohammad S. and Pedersen T. – Combining Lexical and Syntactic Features for Supervised Word Sense Disambiguation**. *In Proceedings of the Conference on Computational Natural Language Learning (CoNLL-2004)*
- **Pakhomov S. – Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts**. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*

References

- **Pakhomov S., Pedersen T. and Chute C. G. – Abbreviation and Acronym Disambiguation in Clinical Discourse**. *To appear in the proceedings of the 2005 Annual Symposium of the American Medical Informatics Association (AMIA 2005)*
- **Pedersen T. – A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation**. *In Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-00)*
- **Vapnik V.: The Nature of Statistical Learning Theory**. *Springer. (1995)*

Software

- **General Architecture for Text Engineering (GATE)**: http://gate.ac.uk/. Cunningham H., Maynard D., Bontcheva K., Tablan V. – GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002)
- **Ngram Statistics Package (NSP)**: http://ngram.sourceforge.net/. Banerjee S. and Pedersen T.: The Design, Implementation and Use of the Ngram Statistics Package. Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (2003)
- **Unstructured Information Management Architecture (UIMA)**: http://www.research.ibm.com/UIMA/. Ferrucci D. and Lally A. – UIMA: an architectural approach to unstructured information processing in the corporate research environment, Natural Language Engineering 10 (2004)
- **Waikato Environment for Knowledge Analysis (WEKA)**: http://www.cs.waikato.ac.nz/ml/weka/. Witten I. and Frank E.: Data Mining: Practical machine learning tools with Java implementation. Morgan-Kaufmann (2000)