Language Processing for Different Domains and Genres: Introduction

Caroline Sporleder, Ines Rehbein

Universität des Saarlandes

Wintersemester 2009/10

15.10.2009

Motivation (1)

Current trends in NLP

- statistical systems vs. rule-based systems
- availability of manually annotated corpora for testing and training systems
- development of state-of-the art NLP tools (part-of-speech taggers, parsers, named entity taggers, semantic role labellers, word sense disambiguators)

Motivation (2)

But . . .

- most annotated data from the news domain or even one specific newspaper (i.e., Wall Street Journal)
- What if you want to process fiction texts, weblogs or scientific papers?

Motivation (2)

But . . .

- most annotated data from the news domain or even one specific newspaper (i.e., Wall Street Journal)
- What if you want to process fiction texts, weblogs or scientific papers?

  ⇒ portability of tools is a problem!

Motivation (2)

But . . .

- most annotated data from the news domain or even one specific newspaper (i.e., Wall Street Journal)
- What if you want to process fiction texts, weblogs or scientific papers?

  $\Rightarrow$ portability of tools is a problem!

  $\Rightarrow$ domain adaptation is a hot research topic

Scope of the Course

1. learn about different domains and genres and their influence on the linguistic properties of a text
2. learn about domain adaptation methods (e.g., data-driven vs. algorithmic)
3. familiarise yourself with the use of different NLP tools

Organisational Stuff

Project Seminar

- for B.Sc. and M.Sc.(CL, LT)
- 5 CPs
- presentation, practical work, report

Seminar

- for B.Sc. and M.Sc. (CL, LT)
- 4 CPs (presentation only), 7 CPs (presentation and term paper)
- presentation, optionally term paper

Course Structure

Mix of practical and theoretical sessions

- weeks 1-5: practical sessions, hands-on experience with NLP tools, tutorials (tutor: Linlin Li)
- from week 6: theoretical part (plus practical work on domain adaptation for project seminar)

Schedule for the first weeks (preliminary)

date	Tut/Sem	What
15.10.	sem	introduction
22.10.	tut	presentation of two pos-taggers (stanford, treetagger) and parsers (stanford, berkeley), exercises
29.10.	tut	presentation of WSD tool, exercises
5.11.	tut	visualisation, machine learning
12.11.	sem	introduction to domains / genres
19.11.	sem	linguistic differences of domains and genres
26.11.	sem	methods for domain adaption

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	$\longrightarrow$	training	$\longrightarrow$	apply to new text

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	$\longrightarrow$	training	$\longrightarrow$	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	$\longrightarrow$	$\boxed{\text{training}}$	$\longrightarrow$	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    → TüBa-D/Z 74% noun attachment
    → TiGer: only 57% noun attachment

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	$\longrightarrow$	$\boxed{\text{training}}$	$\longrightarrow$	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    $\rightarrow$ TüBa-D/Z 74% noun attachment
    $\rightarrow$ TiGer: only 57% noun attachment
    $\Rightarrow$ parsers trained on TüBa-D/Z overgenerate to noun attachment

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	$\rightarrow$	$\boxed{\text{training}}$	$\rightarrow$	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    $\rightarrow$ TüBa-D/Z 74% noun attachment
    $\rightarrow$ TiGer: only 57% noun attachment
    $\Rightarrow$ parsers trained on TüBa-D/Z overgenerate to noun attachment
- Solution:
  - use TüBa-D/Z-trained parsers to parse text from the TüBa-D/Z corpus only?

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	$\rightarrow$	$\boxed{\text{training}}$	$\rightarrow$	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    $\rightarrow$ TüBa-D/Z 74% noun attachment
    $\rightarrow$ TiGer: only 57% noun attachment
    $\Rightarrow$ parsers trained on TüBa-D/Z overgenerate to noun attachment
- Solution:
  - use TüBa-D/Z-trained parsers to parse text from the TüBa-D/Z corpus only? Not a good idea!

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	→	training	→	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    → TüBa-D/Z 74% noun attachment
    → TiGer: only 57% noun attachment
    ⇒ parsers trained on TüBa-D/Z overgenerate to noun attachment
- Solution:
  - use TüBa-D/Z-trained parsers to parse text from the TüBa-D/Z corpus only? Not a good idea!
  - Annotate more data?

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus	→	$\boxed{\text{training}}$	→	apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    → TüBa-D/Z 74% noun attachment
    → TiGer: only 57% noun attachment
    ⇒ parsers trained on TüBa-D/Z overgenerate to noun attachment
- Solution:
  - use TüBa-D/Z-trained parsers to parse text from the TüBa-D/Z corpus only? Not a good idea!
  - Annotate more data? Not feasible!

Domain Adaptation – Why do we need it?

Data-Driven Approaches to NLP

manually annotated corpus $\rightarrow$ $\boxed{\text{training}}$ $\rightarrow$ apply to new text

- Problem:
  - Overfitting (model too closely adapted to training data)
    e.g. distribution of PP attachment in treebanks
    She saw the man/NN (PP with the telescope)
    $\rightarrow$ TüBa-D/Z 74% noun attachment
    $\rightarrow$ TiGer: only 57% noun attachment
    $\Rightarrow$ parsers trained on TüBa-D/Z overgenerate to noun attachment
- Solution:
  - use TüBa-D/Z-trained parsers to parse text from the TüBa-D/Z corpus only? Not a good idea!
  - Annotate more data? Not feasible!
  - Adapt existing tools to new genres and domains

Algorithmic vs. Data-driven – What's the difference?

- Algorithmic
  - Change/improve Machine Learning algorithm to get better performance on new domain
    e.g.: let the algorithm learn the relative importance of features for a specific domain

Domain Adaptation – Algorithmic vs. Data-driven

Algorithmic vs. Data-driven – What's the difference?

- Algorithmic
  - Change/improve Machine Learning algorithm to get better performance on new domain
    e.g.: let the algorithm learn the relative importance of features for a specific domain
- Data-driven
  - Change training data – add training instances from new domain
    e.g.: Active Learning (minimise human annotation effort by carefully selecting the most informative training instances)

Domain Dependence – Why does performance drop?

- $P(x)$ Different distribution in training and test data
  - e.g. Word Sense Disambiguation (WSD): bank (financial institute; Wall Street Journal) vs. bank (river bank; travel guide)

Domain Dependence – Why does performance drop?

- $P(x)$ Different distribution in training and test data
  - e.g. Word Sense Disambiguation (WSD): bank (financial institute; Wall Street Journal) vs. bank (river bank; travel guide)
- $P(y|x)$ same instance has different labels in training and test data
  - She wanted a pet and her parents bought her a mouse.
    She got a new computer and her parents bought her a mouse.

Domain Dependence – Why does performance drop?

- $P(x)$ Different distribution in training and test data
  - e.g. Word Sense Disambiguation (WSD): bank (financial institute; Wall Street Journal) vs. bank (river bank; travel guide)
- $P(y|x)$ same instance has different labels in training and test data
  - She wanted a pet and her parents bought her a mouse.
    She got a new computer and her parents bought her a mouse.
- No training instances for test data
  - e.g. specialised uses/technical terms

Domain Dependence – Why does performance drop?

- $P(x)$ Different distribution in training and test data
  - e.g. Word Sense Disambiguation (WSD): bank (financial institute; Wall Street Journal) vs. bank (river bank; travel guide)
- $P(y|x)$ same instance has different labels in training and test data
  - She wanted a pet and her parents bought her a mouse.
    She got a new computer and her parents bought her a mouse.
- No training instances for test data
  - e.g. specialised uses/technical terms
- Problems caused by unseen words from new domains

Domain Dependence – Why does performance drop?

- $P(x)$ Different distribution in training and test data
  - e.g. Word Sense Disambiguation (WSD): bank (financial institute; Wall Street Journal) vs. bank (river bank; travel guide)
- $P(y|x)$ same instance has different labels in training and test data
  - She wanted a pet and her parents bought her a mouse.
    She got a new computer and her parents bought her a mouse.
- No training instances for test data
  - e.g. specialised uses/technical terms
- Problems caused by unseen words from new domains

Possible solutions?

Domain Adaptation – Possible Approaches

- Algorithmic:
  - Adapt the weights of training instances (some instances generalise to all domains, some are highly domain-specific)

Domain Adaptation – Possible Approaches

- Algorithmic:
  - Adapt the weights of training instances (some instances generalise to all domains, some are highly domain-specific)
  - Adapt feature weights (different weights for features from source/target domain)

Domain Adaptation – Possible Approaches

- Algorithmic:
  - Adapt the weights of training instances (some instances generalise to all domains, some are highly domain-specific)
  - Adapt feature weights (different weights for features from source/target domain)
- Data-driven:
  - Add new training instances from the target domain (human annotation, expensive)

Domain Adaptation – Possible Approaches

- Algorithmic:
  - Adapt the weights of training instances (some instances generalise to all domains, some are highly domain-specific)
  - Adapt feature weights (different weights for features from source/target domain)
- Data-driven:
  - Add new training instances from the target domain (human annotation, expensive)
  - Minimise annotation effort through Active Learning

Domain Adaptation – Possible Approaches

- Algorithmic:
  - Adapt the weights of training instances (some instances generalise to all domains, some are highly domain-specific)
  - Adapt feature weights (different weights for features from source/target domain)
- Data-driven:
  - Add new training instances from the target domain (human annotation, expensive)
  - Minimise annotation effort through Active Learning
  - Semi-supervised approaches, self-training (not clear if it works)

Domain Adaptation – Summary

- Domain Adaptation is an important problem for NLP
- Different approaches/strategies to tackle the problem
  - ML algorithms
  - training data
  - semi-supervised approaches, self-training, re-ranking (?)
  - ...
- There's still a lot to do...

References

- Introduction to Domain Adaptation, Ming-Wei Chang, slides (pdf)
- Jiang, J. and C. Zhai. 2007. Instance weighting for domain adaptation in NLP. In Proc. of the Annual Meeting of the ACL, pp. 264271. pdf
- Domain Adaptation with Structural Correspondence Learning J. Blitzer and R. McDonald and F. Pereira Empirical Methods in Natural Language Processing (EMNLP), 2006 pdf
- Daumé III, H. 2007. Frustratingly easy domain adaptation. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. pdf