An Unsupervised Approach to Domain-Specific Term Extraction

Su Nam Kim†, Timothy Baldwin†, Min-Yen Kan‡

University of Melbourne†
National University of Singapore‡

ALTA Workshop 2009

Introduction

- **Goal**: Automatically extract domain-specific terms (DSWs)
- **Applications**
  - ⋆ keyphrase extraction (Frank et al. 1999, Witten et al. 1999)
  - ⋆ word sense disambiguation (Magnini et al. 2002)
  - ⋆ query expansion and cross-lingual text categorization (Rigutini et al. 2005)
- **Motivation**: the more often a term occurs in particular domain(s), the more likely it is to be domain specific

Related Work

- **Rigutini et al. (2006)** sense-based – accumulate DSWs starting with a seed set, using a thesaurus and sense similarity
- **Kida et al. (2007)** statistical – using web data, collect terms with domain-specificity via technical documents in a given domain
- **Drouin (2004)** statistical – extract unigrams based on their "hypergeometric" distribution
- **Park etal (2008)** statistical – unsupervised, using term frequencies in domains

Unsupervised Domain-Specific Term Extraction: Proposed Method

- **Idea**: similar to TF-IDF, but TF across domains rather than documents

$$
\begin{aligned}
TF_{ij} &= \frac{n_{ij}}{\sum_k n_{kj}} \\
IDF_i &= log(\frac{|D|}{|\{d : t_i \in d\}|}) \\
\textbf{D1} &= TF \cdot IDF_{ij} = TF_{ij} \times IDF_i
\end{aligned}
$$

Unsupervised Domain-Specific Term Extraction: Comparator Method

- **Idea:** directly compare TF in documents for a given domain d with TF in the general document collection

$$\textbf{D2} \quad = \quad domain\_specificity(w) = \frac{\frac{c_d(w)}{N_d}}{\frac{c_g(w)}{N_g}}$$

Domain-Specific Word Collection

- Data
  - ⋆ Modified Lewis split from Reuters collection
  - ⋆ in 90 categories/domains, 3,019 & 7,771 terms in test & training data
  - ⋆ selected DSWs using 3 thresholds

Domain	D1	D2	Domain	D1	D2	Domain	D1	D2
platinum	132	62	oat	115	49	lumber	77	165
lead	71	105	orange	69	160	hog	61	106
pet-chem	55	246	strategic-metal	50	136	income	49	64
fuel	42	80	alum	37	316	rapeseed	35	13
heat	35	58	tin	33	222	silver	29	99
copper	22	236	wpi	20	87	soy-oil	17	18
zinc	14	50	rubber	13	369	gas	13	122
soy-meal	12	23	meal-feed	12	85			

- Human Verification

  - ⋆ over 23 domains which have at least 5 articles in both test and training data sets
  - ⋆ previous method (Drouin 2004) uses human experts' scores
  - ⋆ three LT graduate students asked to assign "yes" or "no" to extracted keyphrases
  - ⋆ initial basic agreement is 69.61% and 73.04% for D1 and D2, respectively.
  - ⋆ accuracy: 40.59% vs. 36.59 for D1 and D2, respectively => conclude D1 is better

Application: Text Categorization

- Extraction
  - ⋆ feature sets: BoW vs. DSW vs. BoW+DSW
  - ⋆ unigrams used as indexing words
  - ⋆ term weighting: TF vs. TF-IDF
  - ⋆ learner: support vector machine (SVM)
  - ⋆ baseline: BoW with frequency $\geq$ 3 (**.677**) (micro-averaged F-score)

- Results

Type	Cutoff	TF			TF-IDF		
		Precision	Recall	F-score	Precision	Recall	F-score
Baseline (BoW)	F1	.586	.473	.524	.738	.596	.660
	F2	.548	.442	.489	.729	.589	.651
	F3	.591	.477	.528	.757	.612	.677
Domain	1	.600	.485	.536	.657	.531	.587
BoW + DSW	1+F1	.652	.527	.583	.762	.615	**.681**
	1+F2	.633	.512	.566	.757	.612	.677
	1+F3	.648	.523	.579	.762	.615	**.681**

Application: Keyphrase Extraction

- Extraction
  - ⋆ feature set: TF-IDF, first occurrence of the word (KEA)
  - ⋆ feature value: Boolean, TF, TF-IDF
  - ⋆ data analysis & statistics:
    - ∗ from 210 test documents, a total of 1,339 keyphrases (6.38 keyphrases per document)
    - ∗ among them, 911 were simplex keyphrases and 428 were NPs
    - ∗ candidate selection method: Nguyen and Kan (2007) method $\geq$ 750 keyphrases were found including 158 NPs
  - ⋆ learners: naive Bayes (NB), maximum entropy (ME)
  - ⋆ baseline: KEA (micro-averaged F-score = **.249**)

- Results

Type	Learner	Features	Precision	Recall	F-score
KEA	NB	–	.193	.208	.200
	ME	–	.240	.259	.249
KEA+ Domain	NB	Boolean	.197	.213	.204
		TF	.192	.208	.200
		TF-IDF	.189	.205	.197
	ME	Boolean	.250	.270	.260
		TF	.251	.272	.261
		TF-IDF	.257	.278	**.267**

Conclusion

- Proposed an automatic method to extract domain-specific terms based on term and document statistics, using a simple adaptation of TF-IDF

- Attested the reliability of the proposed method compared with benchmark system $=>$ small amount of high-quality DSWs collected, well distributed over all domains

- Demonstrated the utility of DSW in text categorization and keyphrase extraction tasks