Identification of fuzzy and underspecified terms in technical documents:

An experiment with distributional semantics

Émilie Merdy(1 & 2), Juyeon Kang(2), Ludovic Tanguy(1)

CLLE-ERSS(1), Prometil(2)

Overview

Context and Objectives

Context

Technical documents like SPECIFICATIONS, PROCEDURES form a specific genre with linguistic constraints: **consistent, non-ambiguous, complete, singular, conforming**, etc.

Potential risk of ambiguities

technical (e.g. malfunctioning), economic (e.g. over-cost of development), human factoring, ecological in case of accidents

Context - Requirements

Requirements: documents in which technical writers state functional needs for a system development (what the system is expected to do).
e.g. *If..., the system shall send the received configuration to its components.*

Importance of writing high quality requirements

In 2015, according to Chaos report (by the Standish group)

- only 29% of projects were successful
- 50% of the challenged projects are related to the errors from the Requirements Engineering
- 70% of them come from the difficulties of understanding of implicit requirements.

Ambiguities

Ambiguities: related work

Different types of ambiguities related to the quality of the specifications: LEXICAL, STRUCTURAL, SEMANTIC (quantifier, negation)

- Zhang, 1998 (General linguistic context)
- Tjong, 2008
- Berry, 2004
- Guidelines like INCOSE, IEEE/ISO 29148-2011
- Internal authoring guidelines of companies

Lexical ambiguities: types

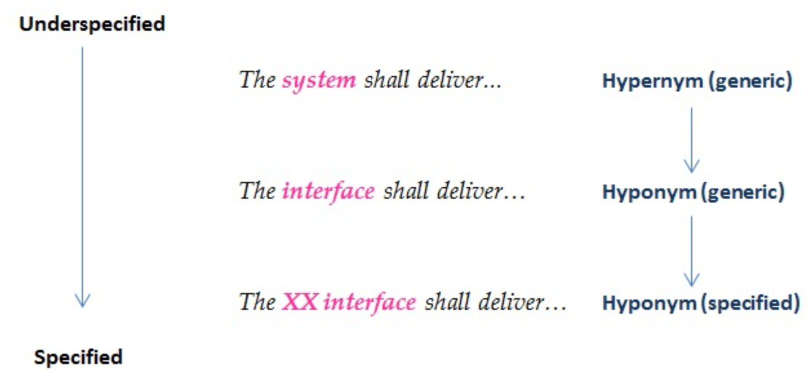
1. **Fuzzy terms**
   - Intrinsically ambiguous: DOMAIN-INDEPENDENT
     e.g. *approximately, nearly, appropriate, ...* (adj, adv (-*ly*))
   - Contextually ambiguous: DOMAIN-DEPENDENT
     e.g. *high, low, maximum, standard ...*

2. **Generic terms (under-specification)**: DOMAIN KNOWLEDGE
   e.g. *system, component, element, manage, request, ...* (noun, verbs)

   *The **components** shall be designed to **operate** [...].*

Lexical ambiguities: examples of generic terms

A requirement should be comprehensible without extra information.
→ by INCOSE (Guide for Writing Requirements)



Lexical ambiguities: examples of fuzzy terms with contextual impact

The ambiguity of an element can be determined by its CONTEXT.

- *The Archiving units shall be able to archive the* ***maximum*** *amount of data.*
- *The* ***maximum*** *pressure loads at the standard operating temperature shall be 6.*

A term can be ambiguous in a context but non-ambiguous in another context. How to deal with it?

Lexical ambiguities: tool and limits

Tool: the example of the SEMIOS system (Lelie project: Kang and Saint-Dizier, 2015)

- automatic detection of potential ambiguities in technical documents (lexical, syntactic, contextual, discursive)
  - lexico-syntactic patterns
  - contextual filtering rules
- important use of lexical resources (open classes, closed classes, business terms, domain specific term), manually updated → Time consuming, increased silence

How to cover new terms when the system needs to be applied to a new domain?

Objectives

Automatic construction of lexical resources: experiments with Automatic Distributional Analysis (ADA)

- determining the nature and the volume of corpora which can show sufficiently relevant semantic relations in technical documents
- extending the lexicons of ambiguous elements (fuzzy terms and generic terms)
- using the extended lexicon (generic terms) to construct specific lexicons from the paradigmatic relations (detected between the hypernyms and its distributional neighbors)

Data

Corpora (1)

1. **Requirements corpus**:
   - five specifications in English (Engine design)
   - 5,186 requirements written in Natural Language
   - limited size of requirements and vocabulary (200,000 tokens, <5,000 types)

Problem

The requirement corpus is extremely small for ADA techniques, so we compare it with a medium, similar domain corpus and a large, less specific corpus to extract different types of semantic neighbors.

Corpora (2)

2. Domain-related Web pages corpus

- Using BootCaT (Baroni & Bernardini, 2004):
  - automatic corpus building method
  - uses a set of terms as seeds: queries for a Web search engine (Bing)
  - collects web pages related to seed terms
  - cleans and PoS-tags the web pages with TreeTagger (Schmidt, 1995)
  - different parameters and customisable filters can make quantity and quality vary

Starting with 51 TECHNICAL TERMS extracted from the requirement corpus with **YaTeA** (Hamon 2012), ended with 2 MILLION TOKENS of web pages.

Corpora (3)

3. **Generic English corpus** UKWaC (Ferraresi *et al.*, 2008)
   - BootCaT process on *.uk* domain webpages *(Baroni & Bernardini, 2004)*, based on generic seeds
   - 2 billion tokens of generic web pages
   - a subset of 200 millions words used for this experiment
   - normalization and PoS-tagging (by TreeTagger)

Basic lexical resources

- Lexical resources from the SEMIOS system, IEEE standard and INCOSE guide
  - Ambiguous terms: *as applicable, always almost, probably, nearly, ...*
  - Generic terms: *the software, the system, malfunction, undesirable effects, ...*

These lexical resources being developed by **linguists** and **domain experts** are considered for this experiment as GOLD REFERENCE.

Automatic Distributional Analysis Application

Principles

- **Hypothesis**: DISTRIBUTIONAL analysis of Harris (Harris, 1954)
- **Used for**: corpus-based unsupervised identification of semantic relations between words

Main idea of distributional analysis :

Two words sharing the same context have a similar meaning.

Example A :
PST phonic wheel rotation speed at 100% : 85000 rpm
PST FREQUENCY detection shall be set on at least 14 consecutive signal period.

Example B :
GTBP phonic wheel rotation speed at 100% : 39978 rpm
GTBP FREQUENCY detection shall be set on at least 25 consecutive signal periods.

Example C :
... CONTROL the temperature / ... LIMIT the temperature
... control the WAIV / ... control the PRSOV

Application of the ADA: Word2Vec (1)

- **Word2vec (Mikolov *et al.*, 2013)**
  - state of the art method (neural embeddings) for distributional analysis
  - distributional proximity between two lexical units, based on common cooccurrences observed in a corpus
  - nature of lexical relation ⟶ UNDER-SPECIFIED but sharing classical semantic relations like SYNONYM, HYPONYM, CO-HYPONYM, ANTONYM

Application of the ADA: Word2vec (2)

Experiments of Word2Vec on our data set:

1. construction of distributional models from three different corpus
   - **pre-processing** of these corpora: TOKENIZED, TAGGED, LEMMATIZED and NORMALIZED as a couple of LEMMA_CATEGORY (e.g. *works* and *worked* ⇒ `work_V`)
   - **parameters**: skip-gram model, six word windows, 200 dimensions
   - **output**: each word represented by a vector, compute cosine similarity between pairs of words

Application of the ADA: Word2Vec (3)

2. Test of distributional models
   - three models generated from the three corpora (requirements, web pages, UKWaC)
   - 16 TERMS pre-selected to test these models (considered as FUZZY by IEEE guideline and our own observations)
   - 8 adjectives: (*easy*, *appropriate*, *best*, *large*, *most*, *normal*, *effective*, *significant*), 3 adverbs: (*about*, *regularly*, *almost*), 5 nouns: (*system*, *malfunction*, *component*, *element*, *software*)
   - for each term, extract the nearest neighbours with the same POS

Application of the ADA: Word2vec (4)

3. Sample results for "MALFUNCTION"

Rank	REQ corpus	Web pages corpus	UKWaC
1	**degradation**	indoor	**harm**
2	fluid	**abnormality**	**interruption**
3	do-160g	thermistor	**delay**
4	**damage**	outdoor	**trouble**
5	service	**failure**	mce

Table: Malfunction's nearest neighbors

Evaluation (1)

THREE EVALUATORS to validate the obtained results (agreement of about 80%)

Criteria for the validation

- Is this neighbor ambiguous in the semantic field of the target word?
- Should it be added to the same lexicon?

	REQ corpus	**Web pages corpus**	**UKWaC**
Adj	6/33	39/126	151/260
Adv	0	8/21	27/71
Nouns(Generic terms)	15/192	24/175	50/256
Total	21/325	71/322	**228/587**

Table: Distribution of relevant neighbors

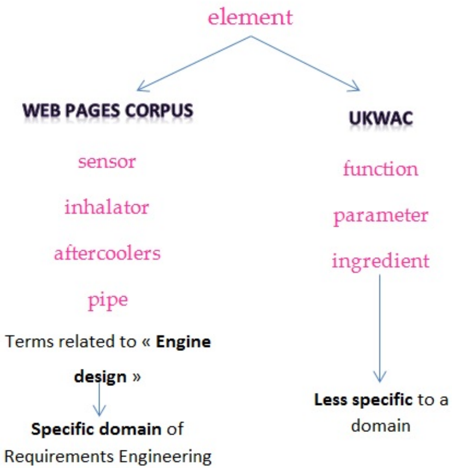
Evaluation (2)

Observations

- The ratio of relevant neighbors largely depends on the CORPUS TYPE AND SIZE.
- The productivity of fuzzy adjectives and adverbs is more important in UKWaC than the other two: because of its SIZE and DOMAIN-INDEPENDENT CHARACTERISTIC.
- Many validated neighbors from the web pages corpus are not found in UKWaC.
- The GENERIC TERMS, validated as relevant neighbors, show different semantic relations:
  - in the UKWaC, SAME DEGREE OF GENERALITY with the pivot (e.g. `system` ⇒ `mechanism, device, tool`)
  - in the Web pages corpus, HYPONYMY RELATIONS with the pivot (e.g. `system` ⇒ `subsystem, unit`)

Evaluation (3)

Example: relevant neighbors of generic term "ELEMENT"



Conclusion and Perspectives

Conclusion

- First experiments to evaluate the relevance of the ADA methods
  - **Objective**: automatic construction of lexical resources in order to help identifying ambiguities in technical documents like requirements
- Tests on the different types and sizes of corpus using Word2vec
  - **small** (200,000 words): Requirements corpus → TOO SMALL to generate interesting results
  - **medium** (2 millions words): Web pages corpus → COMPLEMENTARY WITH UKWAC TO FIND SEMANTIC NEIGHBORS OF GENERIC (UNDER-SPECIFIED) TERMS
  - **large** (200 millions words): UKWaC → 39% OF SEMANTIC NEIGHBORS OF FUZZY TERMS VALIDATED AS RELEVANT
- ADA helps identify SEMANTIC CLASSES (synonym, antonym, hypernym, hyponym and related terms) from the technical documents.

Perspectives

- observations of semantic neighbors of COMPLEX TERMS:
  for example, a fuzzy term **`normal`** in general context, may be unambiguous in aeronautic context when it is used in the complex term **`normal mode`**
- detection of complex noun phrases to identify nouns with its distinctive modifiers (e.g. `xx system`)
- detection of prepositional complements of certain verbs to identify under-specified expressions (e.g. `consider`, `operate`, `provide`, ...used without modifiers)
- use of dependency-based word contexts for narrower distributional similarity (Levy & Goldberg, 2014)

References

Baroni M. and Bernardini S. (2004). Bootcat : Bootstrapping corpora and terms from the web. In LREC.
Berry D.M.and M. Krieger M. (2000). From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity - A Handbook Version 1.0 .
Ferraresi A., Zanchetta E., Baroni M. and Bernardini S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google, p. 47–54.
Harris Z. S. (1954). Distributional structure. Word, 10(2–3), 146–162.
INCOSE (2012). Guide for writing requirements. Rapport interne, International Council on Systems Engineering, requirements working group.
Kang J. and Saint-Dizier P. (2015). Une expérience d'un déploiement industriel de Lelie : une relecture intelligente des exigences. In Actes de INFORSID.
Mikolov T., Chen K., Corrado G. and Dean J. (2013). Efficient estimation of word representations in vector space. In Proceedings of Workshop at ICLR 2013, p. 1–12.
Levy, O and Goldberg, Y. (2014). Dependency-Based Word Embeddings. Proceedings of ACL, Baltimore.
Tjong S. F. (2008). Avoiding ambiguity in requirements specifications. PhD thesis, University of Nottingham.
Zhang Q. (1998). Fuzziness-vagueness-generality-ambiguity. Journal of pragmatics, 29(1), 13–31.