U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Extraction of assembly information from instruction manuals

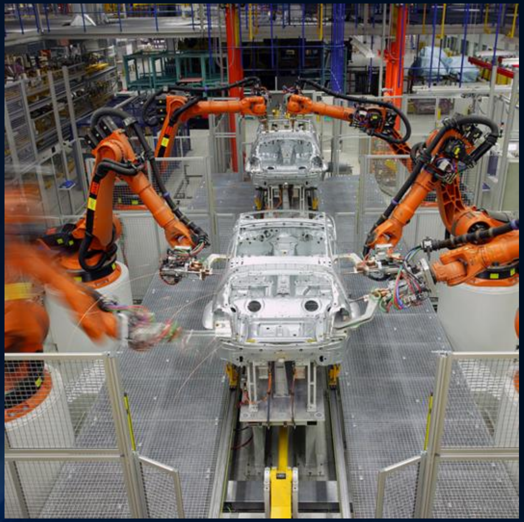
Language Processing and Information Extraction
February, 2017

Carlos Miguel Correia da Costa – carlos.costa@fe.up.pt

Presentation outline

1.a. Context

- Assembly of complex objects by robots requires long programming periods
- Teaching new assembly skills by demonstration can significantly reduce the cost of repurposing robots for building new products
- Automatic extraction of assembly knowledge from instruction manuals can speed up the learning process



Car assembly line

Packaging line

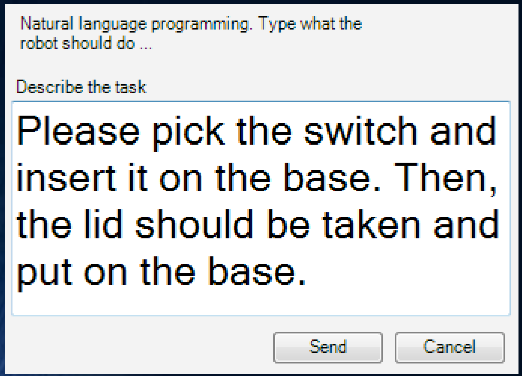
1.b. Objectives

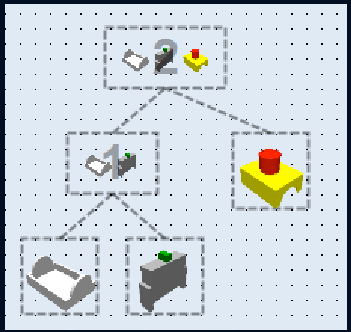
- Development of an information extraction system capable to recognize:
  - The objects being assembled and their parts
  - Tools necessary to perform the assembly
  - The assembly order
  - The spatial relations between the assembly parts

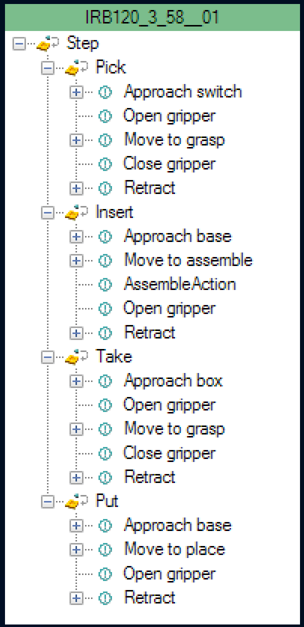
Small parts assembly robot

2.a. Related work

- Extraction of assembly information from written natural language
  - Uses multilingual semantic and syntactic parser and also coreference chains
  - Relies on ontologies for modeling objects, robots, sensors
  - Extraction of predicate-argument structures from text that are associated with robot manipulation skills
  - Identification of assembly objects and spatial constraints
  - Creation of high level symbolic task sequences
  - Generation of code for performing robot arm manipulation







Natural language processing pipeline for assembly of small objects

- Stenmark, M.; Nugues, P., "Natural language programming of industrial robots", in *Robotics (ISR), 2013 44th International Symposium on* , vol., no., pp.1-5, 24-26 Oct. 2013, doi: 10.1109/ISR.2013.6695630

2.b. Related work

- Extraction of tasks from natural language for autonomous robot missions
  - Text in structured format (five paragraphs with [situation], [mission], [execution], [service support] and [command and signals])
  - The goal is to identify named entities useful for mission, such as persons, times, locations, coordinates, targets and organizations
  - Processing pipeline
    - Extraction of plain text from .pdf and .doc files followed by tokenization
    - Usage of Condition Random Fields from jCarafe with the following features:
      - Word lists, regular expressions, prefixes / suffixes, word case, unigram / bigram / trigrams
    - CRF created using 9-fold cross validation
    - NER evaluation using precision (0.702), recall (0.478) and F-measure (0.569)
- D. Chesworth, N. Harmon, L. Tanner, S. Guerlain and M. Balazs, "Named-entity recognition and data visualization techniques to communicate mission command to autonomous systems", *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA, 2016, pp. 233-238. doi: 10.1109/SIEDS.2016.7489305

2.c. Related work

- SpeedRead - Fast Named Entity recognition pipeline
  - Penn Tree bank compliant tokenizer
  - Part-of-Speech (POS) tagger using Condition Random Fields model
  - Knowledge-based named entity recognizer
  - Used F1 score for evaluation
  - Was 10 times faster than the Stanford NER in 2012

- Rami Al-Rfou, Steven Skiena, "SpeedRead: A Fast Named Entity Recognition Pipeline", in 24th International Conference on Computational Linguistics, 2012

2.d. Related work

- Using gazetteers in discriminative information extraction
  - Improvement of gazetteers effectiveness using:
    - Feature normalization
    - Combination of two CRFs using a logarithmic opinion pool approach
      - One CRF trained with and another without gazetteer features
      - Combination approach similar to a mixture model but uses weighted multiplicative combination of models instead of a weighted additive combination

- A. Smith and M. Osborne, "Using gazetteers in discriminative information extraction" in Tenth Conference on Computational Natural Language Learning, 2006

2.e. Related work

- Integrating language, vision and action for human robot dialog systems
  - Multi-agent system capable of learning new skills by analyzing human voice commands, gestures and gaze
  - Speech recognition relied on a Combinatorial Categorial Grammar (CCG) and was used to analyze:
    - Imperative sentences
    - Questions
    - Statements
    - Confirmations
  - Vision system used template matching techniques
- M. Rickert, M. E. Foster, M. Giuliani, T. By, G. Panin, and A. Knoll, "Integrating language, vision and action for human robot dialog systems" in International Conference on Universal Access in Human-Computer Interaction, 2007

2.f. Related work

- Understanding and executing instructions for everyday manipulation tasks from the world wide web
  - Syntactic and sematic analysis
    - Using a Probabilistic Context Free Grammar (PCFG) parser and a Part of Speech (PoS) tagger
  - Word sense retrieval and disambiguation
    - Using a WordNet database and a Cyc ontology
  - Validation of instructions retrieved from the text using a simulator
    - Generation of valid and verified assembly plans
- M. Tenorth, D. Nyga, and M. Beetz, "Understanding and executing instructions for everyday manipulation tasks from the world wide web", in International Conference on Robotics and Automation, 2010

3.a. Dataset text categories

Dataset sources

- Dataset with assembly instructions for 3 different types of objects:
  - Alternators
  - Engines
  - Gearboxes

Engine

Alternator

Gearbox

3.b. Dataset overview

Dataset sources

- Alternators
  - 84 pages
  - 56185 characters
- Engines
  - 148 pages
  - 131287 characters
- Gearboxes
  - 221 pages
  - 192424 characters

4.a. Preprocessing

Dataset preparation

- Extraction of text from pdfs
- Manual verification / correction of text to ensure correct multi-column text extraction
- Selection of assembly procedures
- Removal of pages formatting text
- Creation of validation files with all the assembly parts and their associated quantity for each assembly procedure

4.b. Corpus annotation

- Manual annotation of a representative part of the dataset at the token level (tsv format)
- 9 categories of entities annotated in the corpus
  - Assembly part (PART)
  - Relative position of parts (RPOS)
  - Tools required for assembly (TOOL)
  - Operations performed (OPER)
  - Part IDs (ID)
  - Part quantity (QTY)
  - Part dimension (DIM)
  - Part weight (WGT)
  - Part properties (PROP)
- Creation of train (20370 tokens - 84%) and test (3877 tokens - 16%) corpus

4.b. Corpus annotation

Dataset preparation

	Alternators	Engines	Gearboxes	Global
No of train tokens	4450	9101	6819	20370
No of test tokens	781	1852	1344	3977
No of PART train tokens	738	1709	1477	3924
No of PART test tokens	156	372	327	855
No of RPOS train tokens	83	258	546	887
No of RPOS test tokens	25	52	96	173
No of TOOL train tokens	72	33	33	138
No of TOOL test tokens	5	11	0	16
No of OPER train tokens	182	342	435	959
No of OPER test tokens	30	73	72	175
No of ID train tokens	2	49	76	127
No of ID test tokens	0	35	138	173
No of QTY train tokens	45	41	70	156
No of QTY test tokens	4	22	23	49
No of DIM train tokens	0	67	0	67
No of DIM test tokens	0	5	0	5
No of WGT train tokens	0	2	0	2
No of WGT test tokens	0	0	0	0
No of PROP train tokens	36	63	2	101
No of PROP test tokens	13	2	0	15

Overview of manually annotated dataset at the token level (tsv format)

5.a. CRF training and testing

- Training of the Stanford NER system using the manually annotated train corpus
- Fine tuning of the CRF features by evaluating the precision, recall and F1 metrics of the CRF model when using the testing corpus
  - Starting from the recommended configuration, it was performed 91 tests in which a given feature was activated, deactivated or fine tuned
  - The best configuration achieved an improvement over the recommended features of 3.23 % in F1, 5.79% in recall and 0.35% in precision, resulting in an overall performance of 84.69% in F1, 83.51 in recall and 85.91 in precision

5.b. CRF feature selection

Test folder	Changed Parameter	Recommended Value	Test Value	Train time (sec)	Classifier test Speed (words/sec)	Precision	Recall	F1
0-recommended-properties	-	-	-	125,5	3798,47	0,8556	0,7772	0,8146
93-optimal-configuration	-	-	-	176,4	4581,8	0,8591	0,8351	0,8469
41-lowercaseNGrams-true	lowercaseNGrams	FALSE	TRUE	142,5	3929,84	0,8662	0,7927	0,8278
29-wordShape-chris2	wordShape	chris2useLC	chris2	130	4332,24	0,858	0,7985	0,8272
15-useDisjunctive-false	useDisjunctive	TRUE	FALSE	137,23	4217,39	0,867	0,7859	0,8245
21-wordShape-dan1	wordShape	chris2useLC	dan1	161,1	4389,62	0,8589	0,7927	0,8245
5-noMidNGrams	noMidNGrams	TRUE	FALSE	136,2	3658,69	0,8549	0,7956	0,8242
34-wordFunction-LowercaseAndAmericanizeFunction	wordFunction	null	LowercaseAndAmericanizeFunction	138,3	3980,98	0,8515	0,7965	0,8231
2-useClassFeature-false	useClassFeature	TRUE	FALSE	125	3455,26	0,8617	0,7869	0,8226
22-wordShape-chris1	wordShape	chris2useLC	chris1	159,6	4013,12	0,8568	0,7907	0,8225
18-useTypeSeqs-false	useTypeSeqs	TRUE	FALSE	118,9	4121,24	0,8597	0,7859	0,8212
31-wordShape-chris3	wordShape	chris2useLC	chris3	118,5	4230,85	0,8509	0,7927	0,8208
7-maxNGramLeng-7	maxNGramLeng	6	7	128,6	4058,16	0,8603	0,784	0,8204

Some of the tuning tests that improved the F1 score over the recommended configuration

5.b. CRF feature selection

Test folder	Changed Parameter	Recommended Value	Test Value	Train time (sec)	Classifier test Speed (words/sec)	Precision	Recall	F1
0-recommended-properties	-	-	-	125,5	3798,47	0,8556	0,7772	0,8146
93-optimal-configuration	-	-	-	176,4	4581,8	0,8591	0,8351	0,8469
9-maxNGramLeng-4	maxNGramLeng	6	4	143	4074,8	0,843	0,756	0,7972
80-sigma-5	sigma	1	5	334,9	3751,89	0,8358	0,7608	0,7966
10-maxNGramLeng-3	maxNGramLeng	6	3	128,9	4370,33	0,839	0,7435	0,7883
39-useMoreNeighborNGrams-true	useMoreNeighborNGrams	FALSE	TRUE	141	3933,73	0,8348	0,7454	0,7876
79-sigma-10	sigma	1	10	412,8	4112,72	0,8328	0,7348	0,7807
11-maxNGramLeng-2	maxNGramLeng	6	2	132,6	4199,58	0,8247	0,7213	0,7695
78-sigma-20	sigma	1	20	630,6	4212,92	0,823	0,7175	0,7666
77-sigma-30	sigma	1	30	758,13	3748,35	0,8145	0,7155	0,7618
76-useBagOfWords-true	useBagOfWords	FALSE	TRUE	6362,6	497,56	0,6951	0,8264	0,7551
4-useNGrams-false	useNGrams	TRUE	FALSE	148,9	3827,72	0,8138	0,6914	0,7477
12-maxNGramLeng-1	maxNGramLeng	6	1	138	4177,52	0,8138	0,6914	0,7477

Some of the tuning tests that decreased the F1 score over the recommended configuration

5.c. CRF performance analysis

Entity	Precision	Recall	F1	True positives	False positives	False negatives
DIM	0.0714	0.5000	0.1250	1	13	1
ID	0.9886	0.9158	0.9508	87	1	8
OPER	0.8896	0.8286	0.8580	145	18	30
PART	0.8513	0.7740	0.8108	435	76	127
PROP	0.8889	0.5333	0.6667	8	1	7
QTY	0.3333	0.1724	0.2273	5	10	24
RPOS	0.8777	0.8188	0.8472	122	17	27
TOOL	1.0000	0.3000	0.4615	3	0	7
Totals	0.8556	0.7772	0.8146	806	136	231

NER results using the recommended configuration

5.c. CRF performance analysis   Processing pipeline

Entity	Precision	Recall	F1	True positives	False positives	False negatives
DIM	0.0625	0.5000	0.1111	1	15	1
ID	0.9892	0.9684	0.9787	92	1	3
OPER	0.8935	0.8629	0.8779	151	18	24
PART	0.8667	0.8327	0.8494	468	72	94
PROP	0.6000	0.6000	0.6000	9	6	6
QTY	0.5926	0.5517	0.5714	16	11	13
RPOS	0.8681	0.8389	0.8532	125	19	24
TOOL	1.0000	0.4000	0.5714	4	0	6
Totals	0.8591	0.8351	0.8469	866	142	171

NER results using the fine tuned configuration

5.d. Text annotation

Processing pipeline

- After having a fine tuned CRF model, new unseen text (from the test corpus) was annotated with inline xml tags.

- Example of inline xml text annotation bellow:

- <OPER>**Align**</OPER> <PART>**mounting lugs**</PART> and <OPER>**assemble**</OPER> <QTY>**two**</QTY> halves of <PART>**alternator**</PART> (See Fig. 37), inserting <PART>**rectifier end bearing**</PART> <RPOS>**inner**</RPOS> <PART>**race**</PART> (on <PART>**rotor shaft**</PART>) <RPOS>**into**</RPOS> <PART>**bearing**</PART> in <PART>**rectifier end housing**</PART> (1). <PART>**Seat drive end housing**</PART> (4) at this time. Using <TOOL>**arbor press**</TOOL> and appropriate <PART>**block**</PART> to apply even pressure on end frames.

5.e. Entity extraction

- After having the text annotated, the entities were extracted and split into classes. This allows to quickly see the assembly order of parts.
- Moreover, it is also provided the entity lists sorted and without duplicates (useful to analyze the type of operations or tools needed).

6. Conclusions

- From this project resulted:
  - A new NER dataset for assembly operations in the tsv format
  - A detailed analysis of the features of the Stanford NER system
  - The fine tuning of the Stanford NER system for the new dataset
- The Stanford NER achieved recognition results high enough for being applied in higher level processing systems, such as extraction of assembly operations and learning of new skills by robotic systems

7. Future work

- After having a dataset and a NER system with high recognition performance, future work could include:
  - Generation of the assembly graph for the assembly operations with associated metadata
  - Association of the recognized entities with the CAD models of the assembly parts and tools
  - Semi-supervised learning of the CRF language models for faster deployment of the system to a broad application domain

Thank you!

Questions?