



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JAAKKO PASANEN
NATURAL LANGUAGE UNDERSTANDING FOR INFORMATION
RETRIEVAL IN CHAT BASED CUSTOMER SUPPORT

Master of Science thesis

Examiner: Prof. Ari Visa
Examiner and topic approved by the
Faculty Council of the Faculty of
Engineering Sciences
on 31st December 2016

ABSTRACT

JAAKKO PASANEN: Natural Language Understanding for Information Retrieval
in Chat Based Customer Support

Tampere University of Technology

Master of Science thesis, xx pages, x Appendix pages

December 2016

Master's Degree Programme in Automation Technology

Major: Learning and Intelligent Systems

Examiner: Prof. Ari Visa

Keywords: Hype

The abstract is a concise 1-page description of the work: what was the problem, what was done, and what are the results. Do not include charts or tables in the abstract.

Put the abstract in the primary language of your thesis first and then the translation (when that is needed).

TIIVISTELMÄ

JAAKKO PASANEN: Luonnollisen kielen ymmärrys tiedon hakua varten chat-pohjaisessa asiakaspalvelussa
Tampereen teknillinen yliopisto
Diplomityö, xx sivua, x liitesivua
Joulukuu 2016
Automaatiotekniikan koulutusohjelma
Pääaine: Oppivat ja älykkäät järjestelmät
Tarkastajat: Prof. Ari Visa
Avainsanat: Hype

The abstract in Finnish. Foreign students do not need this page.

Suomenkieliseen diplomityöhön kirjoitetaan tiivistelmä sekä suomeksi että englanniksi.

Kandidaatintyön tiivistelmä kirjoitetaan ainoastaan kerran, samalla kielellä kuin työ. Kuitenkin myös suomenkielisillä kandidaatintöillä pitää olla englanninkielinen otsikko arkistointia varten.

PREFACE

This document template conforms to Guide to Writing a Thesis at Tampere University of Technology (2014) and is based on the previous template. The main purpose is to show how the theses are formatted using LaTeX (or \LaTeX to be extra fancy) .

The thesis text is written into file `d_tyo.tex`, whereas `tutthesis.cls` contains the formatting instructions. Both files include lots of comments (start with `%`) that should help in using LaTeX. TUT specific formatting is done by additional settings on top of the original `report.cls` class file. This example needs few additional files: TUT logo, example figure, example code, as well as example bibliography and its formatting (`.bst`) An example makefile is provided for those preferring command line. You are encouraged to comment your work and to keep the length of lines moderate, e.g. <80 characters. In Emacs, you can use `Alt-Q` to break long lines in a paragraph and `Tab` to indent commands (e.g. inside figure and table environments). Moreover, tex files are well suited for versioning systems, such as Subversion or Git.

Acknowledgements to those who contributed to the thesis are generally presented in the preface. It is not appropriate to criticize anyone in the preface, even though the preface will not affect your grade. The preface must fit on one page. Add the date, after which you have not made any revisions to the text, at the end of the preface.

Tampere, 11.8.2014

On behalf of the working group, Erno Salminen

CONTENTS

0.1	Reading	VII
0.2	Terms for Computational Linguistics	VII
0.3	Universal Dependencies	IX
0.3.1	CoNNL-U format	IX
0.3.2	Universal POS tags	XI
1.	Introduction	1
2.	Natural Language Processing	2
2.1	Pre-processing	2
2.2	Feature Engingeering in NLP	2
2.2.1	Word Embeddings	3
2.2.2	Word2vec	3
2.2.3	GloVe	4
2.2.4	Charater to Word	4
2.3	Annotations	5
2.3.1	Turku Dependency Treebank	6
2.4	Syntactic Parsing	6
2.4.1	POS-tagging	7
2.4.2	Transition Based Parsers	7
2.4.3	Syntaxnet	7
2.5	Co-Reference Parsing	8
2.6	Machine Translation	8
2.7	Lemmatisation	8
2.8	Machine Translation	9
3.	Chat Based Customer Support	10
3.1	Chat Revolution	10
3.2	Automation with Virtual Agents	10
3.3	Existing Systems	10

4. Our System	11
4.1 Pipeline	11
4.1.1 Translating Colloquial Speech	12
4.2 Evaluation	12
5. Customer Business Value and Satisfaction	13
6. Conclusions and Future Work	14
Bibliography	15
APPENDIX A. Something extra	15
APPENDIX B. Something completely different	16

LIST OF ABBREVIATIONS AND SYMBOLS

ANN	Artificial Neural Network
LAS	Labelled Attachment Score
LDA	Latent Dirilecht Allocation
LSA	Latent Semantic Analysis
LSTM	Long short term memory; type of RNN with short term memory.
NER	Named entity recognition
NLP	Natural Language Processing
PMI	Pointwise mutual information
POS	Part-of-speech; also called lexical category
RNN	Recurrent neural network
S-LSTM	Stack long short term memory
TUT	Tampere University of Technology
UAS	Unlabelled Attachment Score

NOTES

0.1 Reading

These vectors can be used as features in a variety of applications, such as information retrieval (Manning et al., 2008), document classification (Sebastiani, 2002), question answering (Tellex et al., 2003). **Pennington2014**

See first paragraph of section 1 of **Liang2016a** for sources on machine translation

Encoder-decoder model

0.2 Terms for Computational Linguistics

1-of-V Coding Representing words as sparse binary vectors which have 1 at the word's vocabulary index and 0 all others. With vocabulary {dog, cat, mouse}, dog becomes [1, 0, 0], cat becomes [0, 1, 0] and mouse [0, 0, 1].

Bag-of-words Multiset of words appearing in a text with occurrence counts for each word. Used as a tool for feature generation. Does not preserve word order or grammar. Can implemented as a dictionary (or associative array) where words are the keys and counts are the values.

Conditional Random Field

Constituent In syntactic analysis, a constituent is a word or a group of words that function(s) as a single unit within a hierarchical structure. Many constituents are phrases. *Yesterday I saw **an orange bird with a white neck***

Corpus A collection of texts with linguistic annotations.

Dimensionality When discussing word embeddings and word vector spaces the dimensionality refers to definition in linear algebra. Dimensionality of arrays in computing means the number of indices required to specify an element in the array. Word vector in 50 dimensional vector space \mathbb{R}^{50} would be represented in computing as one dimensional array of length 50 [d1, d2, d3, ..., d50]

Distributional Hypothesis Words that are used and occur in the same contexts tend to purport similar meanings. **Harris1954**

Feature Numeric data representation that can be effectively exploited in machine learning tasks. E.g. Word occurrence frequencies.

Feature Vector Vector containing all the features. For an image a feature vector could be all the raw values of pixels as a single sequence. For a trigram model with 300 dimensional word embeddings a feature vector would be a 900 dimensional vector formed by concatenating all the separate word embedding vectors.

Gazetteer In Named Entity Recognition a gazetteer is a dictionary of known named entities.

Language Model Probability distribution over sequences of words. Given such a sequence, say of length m , it assigns a probability $P(w_1, \dots, w_m)$ to the whole sequence. Problems caused by growing vocabulary can be addressed with continuous language models such as neural net language models (NNML). Word2Vec by Mikolov2013 addresses this problem with Continuous Bag-of-words and Skip-gram models.

Lemmatisation Process of finding the base form of a word, e.g. flew -> fly

Lexeme A basic lexical unit of a language consisting of one word or several words, the elements of which do not separately convey the meaning of the whole.

n-gram Probabilistic language model where probability of current word is the joint probability of previous n words. Bigram example: $P(I, saw, the, red, house) \approx P(I)P(saw|I)P(red|the)P(house|red)P(\$|house)$. The words unigram, bi-gram and trigram language model denote n-gram model language models with $n = 1$, $n = 2$ and $n = 3$, respectively.

One-hot Group of bits which the legal combinations of values are only those with a single high (1) and all the others low (0). See also 1-of-V Coding.

Parsing Within computational linguistics the term is used to refer to the formal analysis by a computer of a sentence or other string of words into its constituents, resulting in a parse tree showing their syntactic relation to each other, which may also contain semantic and other information.

PMI Word co-occurrence probability metric. High values for words that occur often together.

POS-tagging Process of marking up a word to particular part-of-speech (nouns, verbs, etc...) based on both its definition and its context.

ReLU Rectified Linear Unit. $h(x) = \max\{0, x\}$. Used as non-linear activation function in neural nets particularly in convolutional net

Skip-gram Language model which predicts the context (previous and next n words) of a current word from the current word instead of traditional way of predicting current word from the context.

Structured Prediction Predicting structured objects, rather than scalar discrete or real values. Translating a natural language sentence into a syntactic representation such as a parse tree can be seen as a structured prediction problem in which the structured output domain is the set of all possible parse trees.

Token A structure representing a lexeme that explicitly indicates its categorization for the purpose of parsing. In plain words tokens are instances of words in a text. Not to be confused with word type.

Tree bank Parsed text corpus that annotates syntactic or semantic sentence structure. Contains trees for sentences where phrases in a sentence are structured in a tree of syntactic or semantic relations. Very useful for training POS-taggers etc...

Tri-Training Parsing unlabeled data with two different parsers and selecting only the sentences for which the two parsers produce the same trees **Weiss2015**

Word Lookup Table Matrix $\mathbf{P} \in \mathbb{R}^{d \times |V|}$ of d rows and $|V|$ columns, where d is the word vector dimensionality and $|V|$ is the size of vocabulary. I.e. each column represent a single word and each row represent single dimension in vector space.

Word Type Unique words in a text. *Good wine is good* has 4 tokens but only 3 word types.

Word vector N-dimensional vector representation of a word with interesting properties such as: $\text{vector}(\text{'Paris'}) - \text{vector}(\text{'France'}) + \text{Vector}(\text{'Italy'}) \rightarrow \text{vector}(\text{'Rome'})$

0.3 Universal Dependencies

0.3.1 CoNLL-U format

Universal dependencies use CoNLL-U format for treebanks, CoNLL-U is revised version of CoNLL-X. Annotations are encoded in text files with word lines, blank lines for sentence boundaries and comments starting with hash (#).

Word lines consist of following columns:

ID	Word ID in sentence
FORM	Word form or punctuation symbol
LEMMA	Lemma or stem of word form
UPOSTAG	Universal part-of-speech tag
XPOSTAG	Language specific part-of-speech tag
FEATS	List of morphological features
HEAD	Head of the current token, value of ID or zero (0)
DEPREL	Universal dependency relation to the HEAD
DEPS	List of secondary dependencies
MISC	Any other annotation

Example in Finnish: Jäällä kävely avaa aina hauskoja ja erikoisia näkökulmia kaupunkiin

ID	FORM	LEMMA	UPOSTAG	XPOSTAG
1	Jäällä	jää	NOUN	N
2	kävely	kävely	NOUN	N
3	avaa	avata	VERB	V
4	aina	aina	ADV	Adv
5	hauskoja	hauska	ADJ	A
6	ja	ja	CONJ	C
7	erikoisia	erikoinen	ADJ	A
8	näkökulmia	näkö#kulma	NOUN	N
9	kaupunkiin	kaupunki	NOUN	N
10	.	.	PUNCT	Punct

FEATS

Case=Ade|Number=Sing

Case=Nom|Number=Sing

Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act

—

Case=Par|Degree=Pos|Number=Plur

—

Case=Par|Degree=Pos|Number=Plur

Case=Par|Number=Plur

Case=Ill|Number=Sing

—

HEAD	DEPREL	DEPS	MISC
2	nmod	—	—
3	nsubj	—	—
0	root	—	—
3	advmod	—	—
8	amod	—	—
5	cc	—	—
5	conj	8:amod	—
3	dobj	—	—
8	nmod	—	SpaceAfter=No

0.3.2 Universal POS tags

ADJ	Adjective. Describing word qualifying noun or noun phrase. deep , intelligent
ADP	Adposition. Word expressing spatial or temporal relations under , around , before or mark various semantic roles of , for
ADV	Adverb. Modifies another word. Typically express manner, place, time, frequency etc. She sang loudly . You are quite right.
AUX	Auxiliary verb. A verb used in forming the tenses, moods, and voices of other verbs. Do you want tea?. He has given his all.
CONJ	Coordinating conjunction. Conjunction placed between words, phrases, clauses or sentences of equal rank. and , but , or .
DET	Determiner. Expresses reference of a noun (group). The girl is a student. Which book is that?
INTJ	Interjection. Shows emotion or feeling of the author, includes exclamations, curses, greetings and such. Ouch! , hey , huh? .
NOUN	Noun. Denotes a person, animal, place thing or idea. The cat sat on a mat .
NUM	Numeral. Number, written with digits or letters. 12 , eleven .
PART	Particle. Cannot be inflected. Interjections and conjunctions. In finnish also että , jotta , koska , kun etc...
PRON	Pronoun. Replaces (often previously introduced) noun. Joe saw Jill, and he waved at her .
PUNCT	Punctuation. Full stop, comma, bracket etc.
SCONJ	Subordinating conjunction. A conjunction that introduces a subordinating clause, e.g. although , because , whenever .
SYM	Symbol.
VERB	Verb. Conveys an action bring , read , an occurrence happen , become , or a state of being be , exist .

X Other

1. INTRODUCTION

Testing citation **Andor2016**

2. NATURAL LANGUAGE PROCESSING

- Natural Language Processing is vastly wide field, this thesis discusses only on the sections of NLP relevant to the experiments.
- Subfields such as sentence segmentation and sentiment analysis are out of scope of this thesis.
- Most of the NLP work has been for english.
- Cross-linguistic annotation and parsing has been a reality only after introduction of The Universal Dependencies project and SyntaxNet.
- Similarly Finnish parsing has been unreachable until the first Finnish corpus Turku Dependency Treebank **Haverinen2014** and cross-linguistic parsers.
- Traditionally NLP systems are tailored to the single problem at hand with hand engineered features suited for the problem. Recently general approach has received interest where feature engineering and task specific architectures are not needed. **Collobert2011 Zhang2015**
- This thesis focuses on task specific systems because general language understanding systems are still lacking a good support for production systems and depend on vast amounts of good data **Zhang2015** unobtainable within our time frame and resources.
- See Chapter 2.1 for Finnish Language quirks in **Korenius2004**

2.1 Pre-processing

2.2 Feature Engineering in NLP

- Machine learning algorithms require words to be represented quantifiable features such as IDs or real number vectors.
- Traditional feature selection requires hand engineered features.

- Engineered features hog 95% of the computation time. **Chen2014**
- Traditionally words have been represented by indices. **Mikolov2013**
- Next step was to use 1-of-V coding.
- Index representation is simple as computationally cheap, making use of huge datasets possible. Simple models with huge data outperform complex models with less data. **Mikolov2013**
- See LSA and LDA for previous systems. Neural networks significantly outperform LSA in preserving linearities. LDA doesn't scale for large datasets. **Mikolov2013**

2.2.1 Word Embeddings

- **see section 1.2 of Mikolov2013 for previous work and history of word embeddings**
- Word embeddings represent words as n-dimensional vectors. **Mikolov2013**
- LSA leverages statistical information of a corpus but performs poorly on word analogy task. **Pennington2014**
- Skip-gram is good for word analogies but doesn't utilize corpus statistics well since vectors are trained on local context. **Pennington2014**
- Word embeddings try to map words with semantic similarities close to each other. Words may have several types of similarities such as *France* and *Italy* are countries but *dogs* and *triangles* are both in plural form. **Mikolov2013a**
- **Chen2014** use 50 dimensional word embeddings created with Word2vec.
- **Chen2014** also use embeddings for POS tags and dependency arcs. Only embedding POS tags has clear benefit, **Chen2014** suspect that embedding arc labels have no effect since POS tags already contain the relational information.

2.2.2 Word2vec

- **Mikolov2013**
- Can be used with datasets of billions of words
- Has two models: Continuous bag-of-words and continuous skip-gram

- Continuous bag-of-words predicts current word from the context (surrounding words)
- Continuous skip-gram predicts context (surrounding words) from current word.
- Continuous Bag-of-Words is better for small datasets, continuous skip-gram is better for large datasets.
- CBOW is better for syntax, Skip-gram is better for semantics.
- Can be used to find semantic relationships like $\text{vector('biggest')} - \text{vector('big')} + \text{vector('small')} \Rightarrow \text{vector('smallest')}$
- State of the art (as of 2013)

2.2.3 GloVe

- GloVe by **Pennington2014** capture global corpus statistics with log-bilinear co-occurrence count model.
- Memory requirements for GloVe are substantial since global co-occurrence matrix for entire vocabulary is required, even though GloVe eliminates the need for zero occurrence elements. Problem becomes worse for inflectional languages such as Finnish where vocabulary requires word type for each inflection for each word.
- GloVe outperforms other methods on almost all tested tasks. All tasks are English only. **Pennington2014**

2.2.4 Character to Word

- **Ling2015**
- Word embeddings can be generated from character sequences with significantly better performance for morphological languages.
- Requires only single vector for each character type. Particularly good for morphological languages where word type count may be infinite.
- Orthographical and functional (syntactic and semantic) relationships are non-trivial: *butter* and *batter* are orthographically close but functionally distant, *rich* and *affluent* are orthographically distant but functionally close. **Ling2015** resort to LSTM networks for learning the relationships.

- Word lookup tables are unable to generate representations for previously unseen words, as is required for morphology. **Ling2015**
- C2W can generate embeddings for unseen words.
- C2W is computationally more expensive than word lookup tables, but can be eased by saving word embeddings for most frequent words since the words embedding for a character sequence (word) does not change.
- During training word embeddings change but not inside a single batch, thus it is computationally cheaper to use large batches for training.
- C2W can be replaced with word lookup tables for downstream processing since input and output of both methods are the same.

2.3 Annotations

- Stanford Dependencies by **DeMarneffe2006**
- Stanford Dependencies emerged as de facto annotation scheme for english, but has been adapted to several other languages including Finnish. **Nivre2016**
Haverinen2014
- Turku Dependency Treebank has been tranformed into universal dependencies. **Pyysalo2015**
- Unified annotation scheme reduces need for cross-language adaptations in downstream development. **Petrov2012**
- Universal Dependencies project started from the requirement for cross-linguistically consistent treebank annotations even for morphological languages. **Nivre2016**
- Universal Dependencies project was born from merging several previous attempts to form a cross-linguistically sound dependency annotation schemes. **Nivre2016**
- UD data has been encoded in the CoNLL-U format, a revision of the popular CoNLL-X format. **Nivre2016**
- UD treebanks released in November 2015. **Nivre2016**

2.3.1 Turku Dependency Treebank

- **Haverinen2014**
- Treebanks are needed in computational linguistics.
- First Finnish treebank.
- Open licence, including for text annotated
- 204339 tokens, 15126 sentences
- Based on Stanford Dependency scheme with minor modifications to exclude phenomena not present in Finnish and to include new annotations not present in English.
- Transposed to CoNNL-U scheme by universal dependencies project
- Connexor Machine Syntax is the only currently available Finnish full dependency parser.
- Texts from 10 different categories ranging from news and legal text to blog entries and fiction.
- Dependency parsing is done manually with full double annotation process.
- Uses Omorfi for morphological analysis. Ambiguous tokens are handled partly manually, partly rule based and partly with machine learning.
- FTB uses 3 different taggers for morphology, check them out!
- FTB is 97% grammar examples, meant for rule based POS tagger development

2.4 Syntactic Parsing

- Commonly divided to constituency parsing and dependency parsing.
- Constituency parser creates a parse tree of constituencies.
- Dependency parser creates a parse tree of word token dependencies.
- Constituency parsers are slower but more informal than dependency parsers.
- **Fernandez-Gonzalez2015**
- **Fernandez-Gonzalez2015** show that it is possible to build constituency parser with dependency parser by reducing constituents to dependency parsing.

2.4.1 POS-tagging

- Started from rule based taggers
- Tagger by **Brill1992** (known as Brill tagger) learns the rules and as such can be considered as a hybrid approach
- Contemporary research is focused on statistical and ANN based taggers
- Rest of this section focuses on statistical parsers
- **Ling2015** introduced S-LSTM based State-of-the-art tagger
- **Andor2016** Improved accuracy with transition based tagger
- **Chen2014** were first to represent POS-tag and arc labels as embeddings
- **Andor2016** and **Weiss2015** built their solutions based on **Chen2014**
- **Nivre2004** introduced system for transition based taggers known as arc-standard system. **Chen2014**

2.4.2 Transition Based Parsers

- Good balance between efficiency and accuracy **Weiss2015**
- Parsed left to right; at each position the parser chooses action from a set of possible actions.
- Greedy models are fast but error prone and need hand engineered features **Weiss2015**
- Actions can be chosen by ANN to avoid hand engineering **Chen2014 Weiss2015**

2.4.3 Syntaxnet

- **Andor2016**
- Transition based
- Locally and globally normalized
- Backpropagation through entire net
- State-of-the-Art

2.5 Co-Reference Parsing

2.6 Machine Translation

See section 4.1.1 for more info on our implementation.

2.7 Lemmatisation

- Lemmatization is the process of finding a base form for a word.
- Lemmatization is a normalization technique. **Korenius2004**
- Homographic and inflectional word forms cause ambiguity. **Korenius2004**
- Compound words cause problems. **Korenius2004**
- Lemmatization is better than stemming for clustering of documents written in Finnish because of it's highly inflectional nature. **Korenius2004**
- Lemmatization catches better the semantic meaning of a word, as can be deducted from a better clustering performance.
- Omorfi does lemmatization based on morphological analysis
- Omorfi produces multiple lemmas which need to be disambiguated
- Disambiguation can be done with selecting most probable word, given the context, by language model
- **Kestemont2016** try to solve lemmatization as a neural net classification problem, where lemmas are the class labels
- Method of **Kestemont2016** cannot produce lemmas not seen on training time. This restriction also applies to Word2vec by **Mikolov2013**
- Lemmatization has received a lot of research attention for highly inflectional languages, see **Kestermont2016**
- Unknown token can be taken as a variation of known token, search with e.g. Levenshtein edit distance
- There has been almost none previous work using deep learning for lemmatization before **Kestemont2016**

2.8 Machine Translation

- See section 2 of **Kestemont2016** for sources on Internet to standard language transliteration
- NLP tools suffer with texts with a lot of ortographical variation, one solution is to translate them. **Kestemont2016**

3. CHAT BASED CUSTOMER SUPPORT

Customer support problem domain. History?

3.1 Chat Revolution

3.2 Automation with Virtual Agents

3.3 Existing Systems

4. OUR SYSTEM

- This thesis only considers customer message understanding.
- Greetings, Chatting, information retrieval and response generation are important part of a dialogue system but have been left out of scope of this thesis.
- Large portion of the research on the conversational models focus on chatter systems where utterance responses are retrieved or generated from response corpus. These systems do not aim to solve customer problem by incorporating relevant information retrieved from a specific knowledge base.
- Utterance systems are not suitable for customer support out side of simple frequently asked questions.
- Furthermore large portion of responses generated by neural conversation models are safe responses e.g. *I don't know* **Li2015**
- Conversational models trained on customer support history incorporate data from long time period and as such data may contain outdated information which cannot be corrected unless sentences containing outdated information are replaced. Finding and replacing such information from customer support dialogues requires reading much of the dialogues by humans and rewriting conversation history; task both time consuming, error prone and expensive.
- Also incorporating new information into such dataset requires manual conversation synthesis where humans write new simulated conversations into a dialogue history used for training the system.
- Other ways for updating old and adding new information are required for real world customer support. Our system draws it's information from a knowledge base with traditional user interface for updating and adding new knowledge.

4.1 Pipeline

1. Spell checking and correction (Omorfi)

2. Machine translation from colloquial speech to standard dialect
3. Co-Reference replacement
4. Named Entity Recognition and replacement
5. POS-tagging (Syntaxnet)
6. Dependency parsing (Syntaxnet)
7. Lemmatization (Omorfi)
8. Character to word with **Ling2015** for word lemmas and POS-tags
9. 1-of-V coding for features
10. Dependencies?
11. Query generation
12. Querying knowledge base with generated query
13. Response rendering with templates (sprintf?)

4.1.1 Translating Colloquial Speech

Mä -> Minä

4.2 Evaluation

- Recently end-to-end dialogue system have adopted metrics from machine translation and text summarization. These don't work so well. **Liu2016**
- **Liu2016** considers unsupervised utterance systems where response is generated or selected from a set of possible responses. This is irrelevant since we are building a problem solving dialogue system instead of chattering system.

5. CUSTOMER BUSINESS VALUE AND SATISFACTION

6. CONCLUSIONS AND FUTURE WORK

APPENDIX A. SOMETHING EXTRA

Appendices are purely optional. All appendices must be referred to in the body text

APPENDIX B. SOMETHING COMPLETELY DIFFERENT

You can append to your thesis, for example, lengthy mathematical derivations, an important algorithm in a programming language, input and output listings, an extract of a standard relating to your thesis, a user manual, empirical knowledge produced while preparing the thesis, the results of a survey, lists, pictures, drawings, maps, complex charts (conceptual schema, circuit diagrams, structure charts) and so on.