Russian Morphological Processing for ICALL

Markus Dickinson and Joshua Herring

Dept. of Linguistics, Indiana University

ACL Workshop on Building Educational Applications
Columbus, OH
June 19, 2008

Introduction & Motivation

Intelligent computer-aided language learning (ICALL) systems are ideal for language pedagogy

- provide additional practice outside classroom
  - aiding awareness of language forms & rules (see Amaral and Meurers 2006)

Introduction & Motivation

Intelligent computer-aided language learning (ICALL) systems are ideal for language pedagogy

- provide additional practice outside classroom
  - aiding awareness of language forms & rules (see Amaral and Meurers 2006)

However:

- Few ICALL systems in existence today
  - German (Heift and Nicholson 2001)
  - Portuguese (Amaral and Meurers 2006, 2007)
  - Japanese (Nagata 1995)

Introduction & Motivation

Intelligent computer-aided language learning (ICALL) systems are ideal for language pedagogy

- provide additional practice outside classroom
  - aiding awareness of language forms & rules (see Amaral and Meurers 2006)

However:

- Few ICALL systems in existence today
  - German (Heift and Nicholson 2001)
  - Portuguese (Amaral and Meurers 2006, 2007)
  - Japanese (Nagata 1995)
- Processing of ill-formed learner text focuses on a limited set of languages and language types
  - See Vandeventer Faltin (2003) and references therein

Introduction & Motivation

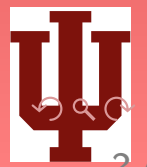
Intelligent computer-aided language learning (ICALL) systems are ideal for language pedagogy

- provide additional practice outside classroom
  - aiding awareness of language forms & rules (see Amaral and Meurers 2006)

However:

- Few ICALL systems in existence today
  - German (Heift and Nicholson 2001)
  - Portuguese (Amaral and Meurers 2006, 2007)
  - Japanese (Nagata 1995)
- Processing of ill-formed learner text focuses on a limited set of languages and language types
  - See Vandeventer Faltin (2003) and references therein

$\Rightarrow$ Should expand to more language families



Re-usability

Significant overhead in developing an ICALL system

Re-usability

Significant overhead in developing an ICALL system

Effort in producing an ICALL system can be reduced by:

- reusing system architecture
  - evaluating and optimizing the architecture

Re-usability

Significant overhead in developing an ICALL system

Effort in producing an ICALL system can be reduced by:

- reusing system architecture
  - evaluating and optimizing the architecture
- adapting existing NLP tools
  - and/or developing resource-light technology

Re-usability

Significant overhead in developing an ICALL system

Effort in producing an ICALL system can be reduced by:

- reusing system architecture
  - evaluating and optimizing the architecture
- adapting existing NLP tools
  - and/or developing resource-light technology

It is important to determine where and how reuse of technology is appropriate

Russian ICALL

We are developing an ICALL system for beginning learners of Russian

Russian ICALL

We are developing an ICALL system for beginning learners of Russian

- Based on the TAGARELA system for Portuguese (Amaral and Meurers 2006, 2007)
  - **Q1:** How can the technology in TAGARELA can be adapted for efficient & accurate use with Russian?

Russian ICALL

We are developing an ICALL system for beginning learners of Russian

- Based on the TAGARELA system for Portuguese (Amaral and Meurers 2006, 2007)
  - **Q1:** How can the technology in TAGARELA can be adapted for efficient & accurate use with Russian?
- Requires development of techniques to parse ill-formed input for a morphologically-rich language
  - **Q2:** What kind of processing do we need, and are existing NLP tools reusable for this purpose?

Russian ICALL

We are developing an ICALL system for beginning learners of Russian

- Based on the TAGARELA system for Portuguese (Amaral and Meurers 2006, 2007)
  - **Q1:** How can the technology in TAGARELA can be adapted for efficient & accurate use with Russian?
- Requires development of techniques to parse ill-formed input for a morphologically-rich language
  - **Q2:** What kind of processing do we need, and are existing NLP tools reusable for this purpose?
    - **Q2a:** What is the context for processing (i.e., the exercise requirements)?

Russian ICALL

We are developing an ICALL system for beginning learners of Russian

- Based on the TAGARELA system for Portuguese (Amaral and Meurers 2006, 2007)
  - **Q1:** How can the technology in TAGARELA can be adapted for efficient & accurate use with Russian?
- Requires development of techniques to parse ill-formed input for a morphologically-rich language
  - **Q2:** What kind of processing do we need, and are existing NLP tools reusable for this purpose?
    - **Q2a:** What is the context for processing (i.e., the exercise requirements)?
    - **Q2b:** What are the expected types of morphological errors?

System architecture

From TAGARELA, we retain:

- Modular separation of activities from analysis
  - Each activity type has own directory, to ease in:
    - loading different kinds of external files (e.g., sound)
    - calling different processing tools (Amaral 2007)

System architecture

From TAGARELA, we retain:

- Modular separation of activities from analysis
  - Each activity type has own directory, to ease in:
    - loading different kinds of external files (e.g., sound)
    - calling different processing tools (Amaral 2007)
- Web processing code
  - e.g., code for handling user logins, design of user databases (for tracking learner information)
  - Minimizes amount of online overhead in our system, allowing us to focus on linguistic processing

System architecture

From TAGARELA, we retain:

- Modular separation of activities from analysis
  - Each activity type has own directory, to ease in:
    - loading different kinds of external files (e.g., sound)
    - calling different processing tools (Amaral 2007)
- Web processing code
  - e.g., code for handling user logins, design of user databases (for tracking learner information)
  - Minimizes amount of online overhead in our system, allowing us to focus on linguistic processing
- Idea of using annotation-based processing (cf. Amaral and Meurers 2007).
  - Before error detection/diagnosis, annotate learner input with linguistic properties that can be automatically determined

Exercise design

Goals of the system:

- Support an 8-week "survival" Russian course
  - Basics of the language
  - Contextualized practice to support traveling to Russia

Exercise design

Goals of the system:

- Support an 8-week "survival" Russian course
  - Basics of the language
  - Contextualized practice to support traveling to Russia
- Cover a range of exercises, all of which require some morphosyntactic analysis of Russian
  - listening, video-based narratives, reading practice, exercises centered around maps and locations, ...

Exercise design

Goals of the system:

- Support an 8-week "survival" Russian course
  - Basics of the language
  - Contextualized practice to support traveling to Russia
- Cover a range of exercises, all of which require some morphosyntactic analysis of Russian
  - listening, video-based narratives, reading practice, exercises centered around maps and locations, ...

A simple example of a Russian verbal exercise:

(1) Вчера он __ (видеть) фильм.
vchera on __ (videt') fil'm
Yesterday he __ (to see) a film

Exercise design

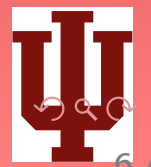
Goals of the system:

- Support an 8-week "survival" Russian course
  - Basics of the language
  - Contextualized practice to support traveling to Russia
- Cover a range of exercises, all of which require some morphosyntactic analysis of Russian
  - listening, video-based narratives, reading practice, exercises centered around maps and locations, ...

A simple example of a Russian verbal exercise:

(1) Вчера он __ (видеть) фильм.
vchera on __ (videt') fil'm
Yesterday he __ (to see) a film

⇒ This set-up constrains what types of errors learners are allowed to make



Expected error types (1)

We focus on morphological errors, as these are common across exercises

Expected error types (1)

We focus on morphological errors, as these are common across exercises

1. Inappropriate verb stem

Expected error types (1)

We focus on morphological errors, as these are common across exercises

1. Inappropriate verb stem
   1.1 Always inappropriate (spelling error)
   - Requires some spell-checking technology

Expected error types (1)

We focus on morphological errors, as these are common across exercises

1. Inappropriate verb stem
   - 1.1 Always inappropriate (spelling error)
     - Requires some spell-checking technology
   - 1.2 Inappropriate for this context
     - Requires activity model specifying appropriate verbs

Expected error types (1)

We focus on morphological errors, as these are common across exercises

1. Inappropriate verb stem
   - 1.1 Always inappropriate (spelling error)
     - Requires some spell-checking technology
   - 1.2 Inappropriate for this context
     - Requires activity model specifying appropriate verbs

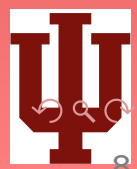
External needs: lexicon, spell checker

Expected error types (2)

2. Inappropriate verb affix

Expected error types (2)

2. Inappropriate verb affix
   2.1 Always inappropriate (spelling error)



Expected error types (2)

2. Inappropriate verb affix
   - 2.1 Always inappropriate (spelling error)
   - 2.2 Always inappropriate for verbs
     - ев is an appropriate nominal ending:

(2) *начина-ев
begin-??

Expected error types (2)

2. Inappropriate verb affix
   - 2.1 Always inappropriate (spelling error)
   - 2.2 Always inappropriate for verbs
     - ев is an appropriate nominal ending:

       (2) *начина-ев
       begin-??

   - 2.3 Inappropriate for this verb
     - ит is for a different verb conjugation:

       (3) *начина-ит (cf. начина-ет)
       begin-3S

Expected error types (2)

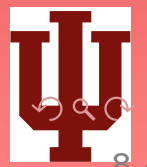
2. Inappropriate verb affix
   - 2.1 Always inappropriate (spelling error)
   - 2.2 Always inappropriate for verbs
     - ев is an appropriate nominal ending:

       (2) *начина-ев
       begin-??

   - 2.3 Inappropriate for this verb
     - ит is for a different verb conjugation:

       (3) *начина-ит (cf. начина-ет)
       begin-3S

External needs: lexicon, spell checker



Expected error types (3)

3. Inappropriate combination of stem and affix
   - The verb for 'can' varies between the stems мог and мож (e.g., мож-ем 'we can')

     (4) *мож-у (cf. мог-у)
         can-1s

Expected error types (3)

3. Inappropriate combination of stem and affix
   - The verb for 'can' varies between the stems мог and мож (e.g., мож-ем 'we can')

   (4) *мож-у (cf. мог-у)
   can-1s

External needs: lexicon

Expected error types (4)

4. Well-formed word in inappropriate context

Expected error types (4)

4. Well-formed word in inappropriate context
   4.1 Inappropriate agreement features
   - Need to know best analysis in context of verb & subject

(5) *Я думает
I think-3SG

Expected error types (4)

4. Well-formed word in inappropriate context
   - 4.1 Inappropriate agreement features
     - Need to know best analysis in context of verb & subject

     (5) *Я думает
     I think-3SG

   - 4.2 Inappropriate verb form (tense, (im)perfective, etc.)
     - Activity model can often indicate correct form—e.g., perfective (completed action) or imperfective

Expected error types (4)

4. Well-formed word in inappropriate context
   - 4.1 Inappropriate agreement features
     - Need to know best analysis in context of verb & subject

     (5) *Я думает
         I think-3SG

   - 4.2 Inappropriate verb form (tense, (im)perfective, etc.)
     - Activity model can often indicate correct form—e.g., perfective (completed action) or imperfective
     - Need to know best analysis in context—e.g., infinitive verb is governed by a verb selecting for infinitive

Expected error types (4)

4. Well-formed word in inappropriate context
   - 4.1 Inappropriate agreement features
     - Need to know best analysis in context of verb & subject

     (5) *Я думает
         I think-3SG

   - 4.2 Inappropriate verb form (tense, (im)perfective, etc.)
     - Activity model can often indicate correct form—e.g., perfective (completed action) or imperfective
     - Need to know best analysis in context—e.g., infinitive verb is governed by a verb selecting for infinitive

External needs: morphological analyzer, POS tagger

Using the error taxonomy

Even for simple exercises, there are a range of errors, requiring new technology

Using the error taxonomy

Even for simple exercises, there are a range of errors, requiring new technology

- Error types #1 through #3 make no use of context
  - Only need information from activity model and lexicon to tell whether the word is valid
  - Priority is thus to develop or acquire a lexicon

Using the error taxonomy

Even for simple exercises, there are a range of errors, requiring new technology

- Error types #1 through #3 make no use of context
  - Only need information from activity model and lexicon to tell whether the word is valid
  - Priority is thus to develop or acquire a lexicon
- Error type #4 requires contextual information, as the words are well-formed
  - Requires morphological analysis, based on a lexicon
  - Ideally, the lexicon design should be integrated with morphological analysis

Using the error taxonomy

Even for simple exercises, there are a range of errors, requiring new technology

- Error types #1 through #3 make no use of context
  - Only need information from activity model and lexicon to tell whether the word is valid
  - Priority is thus to develop or acquire a lexicon
- Error type #4 requires contextual information, as the words are well-formed
  - Requires morphological analysis, based on a lexicon
  - Ideally, the lexicon design should be integrated with morphological analysis
- No category for argument structure misuse or word order variation as these are syntactic errors, not morphological

Morphological analysis

Annotation of input must be able to determine morphological properties, independent of surrounding context

Morphological analysis

Annotation of input must be able to determine morphological properties, independent of surrounding context

- We cannot assume well-formed input, as traditional morphological analyzers do

Morphological analysis

Annotation of input must be able to determine morphological properties, independent of surrounding context

- We cannot assume well-formed input, as traditional morphological analyzers do
- We need ready access to alternative analyses, especially for learner innovations

(6) душ-у
soul-N.ACC?
*shower-V.1S?

Morphological analysis

Annotation of input must be able to determine morphological properties, independent of surrounding context

- We cannot assume well-formed input, as traditional morphological analyzers do
- We need ready access to alternative analyses, especially for learner innovations

  (6) душ-у
  soul-N.ACC?
  *shower-V.1S?

- We need easy implementation of activity-specific heuristics, e.g., weight analyses

Morphological analysis

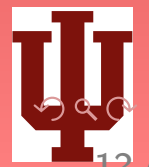
Annotation of input must be able to determine morphological properties, independent of surrounding context

- We cannot assume well-formed input, as traditional morphological analyzers do
- We need ready access to alternative analyses, especially for learner innovations

  (6) душ-у
  soul-N.ACC?
  *shower-V.1S?

- We need easy implementation of activity-specific heuristics, e.g., weight analyses

Finite State Morphology is ideal for this purpose (see, e.g., Roark and Sproat 2007)



The nature of the lexicon

Goal: Accurately obtain partial information from well-formed and ill-formed input

The nature of the lexicon

Goal: Accurately obtain partial information from well-formed and ill-formed input

Proposal: Use a fully-specified lexicon, implemented as a Finite State Transducer (FST), indexed by both word edges

The nature of the lexicon

Goal: Accurately obtain partial information from well-formed and ill-formed input

Proposal: Use a fully-specified lexicon, implemented as a Finite State Transducer (FST), indexed by both word edges

- Russian morphological information is at word edges—i.e., prefixes and suffixes
  - Analysis proceeds by working inwards, one character at a time, beginning at each end of an input item

Lexical chains

Specifically, morphological endings are stored as separate chains, attached to the main chain as appropriate

Lexical chains

Specifically, morphological endings are stored as separate chains, attached to the main chain as appropriate

- Read symbols from input string one at a time, building a set of hypotheses about the proper analysis

Lexical chains

Specifically, morphological endings are stored as separate chains, attached to the main chain as appropriate

- Read symbols from input string one at a time, building a set of hypotheses about the proper analysis
  - set of legal continuations of the current string
  - set of continuations that can be obtained through application of a *repair operation* (insert, delete, etc.)

Lexical chains

Specifically, morphological endings are stored as separate chains, attached to the main chain as appropriate

- Read symbols from input string one at a time, building a set of hypotheses about the proper analysis
  - set of legal continuations of the current string
  - set of continuations that can be obtained through application of a *repair operation* (insert, delete, etc.)

Consider дума-ю ('think-1sg'):

- Up to morpheme boundary, identical to some form of дума (*duma*), 'parliament'

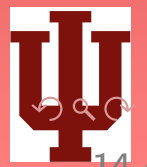
Lexical chains

Specifically, morphological endings are stored as separate chains, attached to the main chain as appropriate

- Read symbols from input string one at a time, building a set of hypotheses about the proper analysis
  - set of legal continuations of the current string
  - set of continuations that can be obtained through application of a *repair operation* (insert, delete, etc.)

Consider дума-ю ('think-1sg'):

- Up to morpheme boundary, identical to some form of дума (*duma*), 'parliament'
- At hypothesized boundary, both competing hypotheses ('think' and 'parliament') are possible
  - For 'think', continuing to ю is legal
  - For 'parliament', continuing to ю requires a repair



Information for feedback

As it changes state, the transducer will add information to the current set of analyses:

Information for feedback

As it changes state, the transducer will add information to the current set of analyses:

- Append input symbol to output

Information for feedback

As it changes state, the transducer will add information to the current set of analyses:

- Append input symbol to output
  - Add morphological features, generally when a transition crosses a morphological boundary

Information for feedback

As it changes state, the transducer will add information to the current set of analyses:

- Append input symbol to output
  - Add morphological features, generally when a transition crosses a morphological boundary
  - Add corrections on the input string, when phonological processes have been misapplied

Information for feedback

As it changes state, the transducer will add information to the current set of analyses:

- Append input symbol to output
  - Add morphological features, generally when a transition crosses a morphological boundary
  - Add corrections on the input string, when phonological processes have been misapplied

Hypothesizing morpheme boundaries means we can:

- segment word into its likely component parts

Information for feedback

As it changes state, the transducer will add information to the current set of analyses:

- Append input symbol to output
  - Add morphological features, generally when a transition crosses a morphological boundary
  - Add corrections on the input string, when phonological processes have been misapplied

Hypothesizing morpheme boundaries means we can:

- segment word into its likely component parts
- analyze each part independently of the others
  - e.g., ignore an erroneous morpheme while identifying an adjoining correct morpheme

Efficiency

Is fully specifying every word wasteful of memory?

Efficiency

Is fully specifying every word wasteful of memory?

- Since the lexicon is an FST, sections shared across forms will only be stored once
  - stems which require such affixes simply point to them

Efficiency

Is fully specifying every word wasteful of memory?

- Since the lexicon is an FST, sections shared across forms will only be stored once
  - stems which require such affixes simply point to them
- Added advantage: analyzer operating over FST lexicon retains explicit knowledge of state

Efficiency

Is fully specifying every word wasteful of memory?

- Since the lexicon is an FST, sections shared across forms will only be stored once
  - stems which require such affixes simply point to them
- Added advantage: analyzer operating over FST lexicon retains explicit knowledge of state
  - easy to entertain competing analyses (Ćavar 2008)

Efficiency

Is fully specifying every word wasteful of memory?

- Since the lexicon is an FST, sections shared across forms will only be stored once
  - stems which require such affixes simply point to them
- Added advantage: analyzer operating over FST lexicon retains explicit knowledge of state
  - easy to entertain competing analyses (Ćavar 2008)
  - easy to return to previous points in an analysis to resolve ambiguities (cf., e.g., Beesley and Karttunen 2003)

Efficiency

Is fully specifying every word wasteful of memory?

- Since the lexicon is an FST, sections shared across forms will only be stored once
  - stems which require such affixes simply point to them
- Added advantage: analyzer operating over FST lexicon retains explicit knowledge of state
  - easy to entertain competing analyses (Ćavar 2008)
  - easy to return to previous points in an analysis to resolve ambiguities (cf., e.g., Beesley and Karttunen 2003)

The error taxonomy prevents all possible paths from being simultaneously entertained

Sketch of error detection

Analyzer will try to build a path based on information it has

Sketch of error detection

Analyzer will try to build a path based on information it has

- Inappropriate ending for a verb

(7) *начина-ев
begin-??

Sketch of error detection

Analyzer will try to build a path based on information it has

- Inappropriate ending for a verb

  (7) *начина-ев
  begin-??

- Analyzers working from both directions will find same morpheme boundary

Sketch of error detection

Analyzer will try to build a path based on information it has

- Inappropriate ending for a verb

  (7) *начина-ев
  begin-??

- Analyzers working from both directions will find same morpheme boundary
- Analysis of начина- and of -ев are easily identified as incompatible

Sketch of error detection

Analyzer will try to build a path based on information it has

- Inappropriate ending for *this* verb

(7) *начина-ит (cf. начина-ет)
begin-3S

Sketch of error detection

Analyzer will try to build a path based on information it has

- Inappropriate ending for *this* verb

  (7) *начина-ит (cf. начина-ет)
  begin-3S

- Analyzers working from both directions will find same morpheme boundary

Sketch of error detection

Analyzer will try to build a path based on information it has

- Inappropriate ending for *this* verb

  (7) *начина-ит (cf. начина-ет)
  begin-3S

- Analyzers working from both directions will find same morpheme boundary
- Analysis of начина- and of -ит do not match in features
  - Morphological information from affix will enable the repair operation substitution to find the right continuation.

Constructing the Lexicon

- Lexicon generation can be done semi-automatically

Constructing the Lexicon

- Lexicon generation can be done semi-automatically
- We need:
  - Freely-available corpus (Sharoff et al. 2008)
  - A handful of inflected forms to derive common morphological paradigms
  - Unsupervised morphology learner like Linguistica (Goldsmith and Hu 2004)

Summary & Outlook

SUMMARY:

- An FST lexicon provides a way to do morphological error analysis on learner language in Russian that is:
  1. easily optimizable for learner environments
  2. accurate without sacrificing generality
  3. flexible enough to detect even unanticipated errors
- We believe this approach is applicable to a number of languages

Summary & Outlook

Summary:

- An FST lexicon provides a way to do morphological error analysis on learner language in Russian that is:
  1. easily optimizable for learner environments
  2. accurate without sacrificing generality
  3. flexible enough to detect even unanticipated errors
- We believe this approach is applicable to a number of languages

Next Steps:

1. Construction of lexicon for small subset of the language relevant to our exercises
2. Performing/testing error detection and diagnosis on top of the linguistic analysis
3. Addition of linguistic analysis beyond the word level, operating in parallel with the morphological analyzer

Acknowledgments

We would like to thank

- Detmar Meurers and Luiz Amaral for providing us with the TAGARELA sourcecode & insights into ICALL systems
- Anna Feldman and Jirka Hana for advice on Russian resources
- Two anonymous reviewers for insightful comments

This research was supported by grant P116S070001 through the U.S. Department of Education's Fund for the Improvement of Postsecondary Education.

References

Amaral, Luiz (2007). Designing Intelligent Language Tutoring Systems: integrating Natural Language Processing technology into foreign language teaching. Ph.D. thesis, The Ohio State University.

Amaral, Luiz and Detmar Meurers (2006). Where does ICALL Fit into Foreign Language Teaching? Talk given at CALICO Conference. University of Hawaii, http://purl.org/net/icall/handouts/calico06-amaral-meurers.pdf.

Amaral, Luiz and Detmar Meurers (2007). Putting activity models in the driver's seat: Towards a demand-driven NLP architecture for ICALL. Talk given at EUROCALL. University of Ulster, Coleraine Campus, http://purl.org/net/icall/handouts/eurocall07-amaral-meurers.pdf.

Beesley, Kenneth R. and Lauri Karttunen (2003). *Finite State Morphology*. CSLI Publications.

Ćavar, Damir (2008). The Croatian Language Repository: Quantitative and Qualitative Resources for Linguistic Research and Language Technologies. Invited talk, Indiana University Department of Lingistics, January 2008.

Clemenceau, David (1997). Finite-State Morphology: Inflections and Derivations in a Singl e Framework Using Dictionaries and Rules. In Emmanuel Roche and Yves Schabes (eds.), *Finite State Language Processing*, The MIT Press.

Goldsmith, John and Yu Hu (2004). From Signatures to Finite State Automata. In *Midwest Computational Linguistics Colloquium (MCLC-04)*. Bloomington, IN.

Heift, Trude and Devlan Nicholson (2001). Web delivery of adaptive and interactive language tutoring. *International Journal of Artificial Intelligence in Education* 12(4), 310–325.

Koskenniemi, Kimmo (1983). Two-level morphology: a general computational model for word-fo rm recognition and production. Ph.D. thesis, University of Helsinki.

Murray, Janet H. (1995). Lessons Learned from the Athena Language Learning Project: Using Natural Language Processing, Graphics, Speech Processing, and Interactive Video for Communication-Based Language Learning. In V. Melissa Holland, Michelle R. Sams and Jonathan D. Kaplan (eds.), *Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Associates, chap. 13, pp. 243–256.

Nagata, Noriko (1995). An Effective Application of Natural Language Processing in Second Language Instruction. *CALICO Journal* 13(1), 47–67.

Roark, Brian and Richard Sproat (2007). *Computational Approaches to Morphology and Syntax*. Oxford University Press.

Sharoff, Serge, Mikhail Kopotev, Tomaž Erjavec, Anna Feldman and Dagmar Divjak (2008). Designing and evaluating Russian tagsets. In *Proceedings of LREC 2008*. Marrakech.

Vandeventer Faltin, Anne (2003). Syntactic error diagnosis in the context of computer assisted language learning. Thèse de doctorat, Université de Genève, Genève.