Semantic analysis & Lexical Semantic
ICS 482 Natural Language Processing

Lecture 22: Semantic analysis & Lexical Semantic

Husni Al-Muhtaseb

بسم الله الرحمن الرحيم

ICS 482 Natural Language Processing

Lecture 22: Semantic analysis & Lexical Semantic

Husni Al-Muhtaseb

NLP Credits and Acknowledgment

These slides were adapted from presentations of the Authors of the book

SPEECH and LANGUAGE PROCESSING:

An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition

and some modifications from presentations found in the WEB by several scholars including the following

NLP Credits and Acknowledgment

If your name is missing please contact me
muhtaseb
At
Kfupm.
Edu.
sa

NLP Credits and Acknowledgment

Husni Al-Muhtaseb
James Martin
Jim Martin
Dan Jurafsky
Sandiway Fong
Song young in
Paula Matuszek
Mary-Angela Papalaskari
Dick Crouch
Tracy Kin
L. Venkata Subramaniam
Martin Volk
Bruce R. Maxim
Jan Hajič
Srinath Srinivasa
Simeon Ntafos
Paolo Pirjanian
Ricardo Vilalta
Tom Lenaerts

Heshaam Feili
Björn Gambäck
Christian Korthals
Thomas G. Dietterich
Devika Subramanian
Duminda Wijesekera
Lee McCluskey
David J. Kriegman
Kathleen McKeown
Michael J. Ciaraldi
David Finkel
Min-Yen Kan
Andreas Geyer-Schulz
Franz J. Kurfess
Tim Finin
Nadjet Bouayad
Kathy McCoy
Hans Uszkoreit
Azadeh Maghsoodi

Khurshid Ahmad
Staffan Larsson
Robert Wilensky
Feiyu Xu
Jakub Piskorski
Rohini Srihari
Mark Sanderson
Andrew Elks
Marc Davis
Ray Larson
Jimmy Lin
Marti Hearst
Andrew McCallum
Nick Kushmerick
Mark Craven
Chia-Hui Chang
Diana Maynard
James Allan

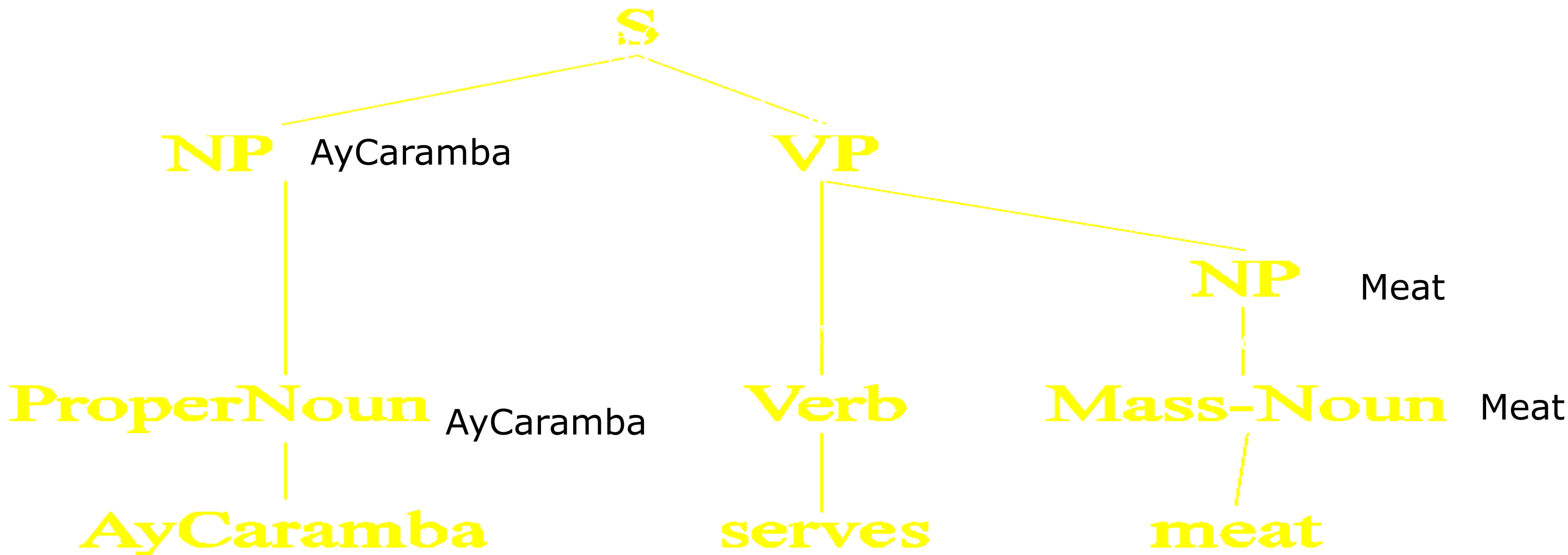
Martha Palmer
julia hirschberg
Elaine Rich
Christof Monz
Bonnie J. Dorr
Nizar Habash
Massimo Poesio
David Goss-Grubbs
Thomas K Harris
John Hutchins
Alexandros Potamianos
Mike Rosner
Latifa Al-Sulaiti
Giorgio Satta
Jerry R. Hobbs
Christopher Manning
Hinrich Schütze
Alexander Gelbukh
Gina-Anne Levow
Guitao Gao
Qing Ma
Zeynep Altan

Previous Lectures

- Introduction and Phases of an NLP system
- NLP Applications - Chatting with Alice
- Finite State Automata & Regular Expressions & languages
- Morphology: Inflectional & Derivational
- Parsing and Finite State Transducers, Porter Stemmer
- Statistical NLP – Language Modeling
- N Grams, Smoothing
- Parts of Speech - Arabic Parts of Speech
- Syntax: Context Free Grammar (CFG) & Parsing
- Parsing: Earley's Algorithm
- Probabilistic Parsing
- Probabilistic CYK - Dependency Grammar
- Semantics: Representing meaning - FOPC
- Lexicons and Morphology – invited lecture
- Semantics: Representing meaning
- Semantic Analysis: Syntactic-Driven Semantic Analysis

Today's Lecture

- Semantic Analysis (~ Ch 15)
  - Syntactic-Driven Semantic Analysis
  - Semantic Grammars
- Presentations
  - Evaluation
  - How to give good presentation



- $S \rightarrow NP\ VP$ — {VP.sem(NP.sem)}
- $VP \rightarrow Verb\ NP$ — {Verb.sem(NP.sem)}
- $Verb \rightarrow serves$ — $\lambda x\, \lambda y\ \exists e\ Serving(e) \wedge Server(e, y) \wedge Served(e, x)$
- $NP \rightarrow PropNoun$ — {PropNoun.sem}
- $NP \rightarrow MassNoun$ — {MassNoun.sem}
- PropNoun → *AyCaramba* — {AyCaramba}
- MassNoun → *meat* — {MEAT}

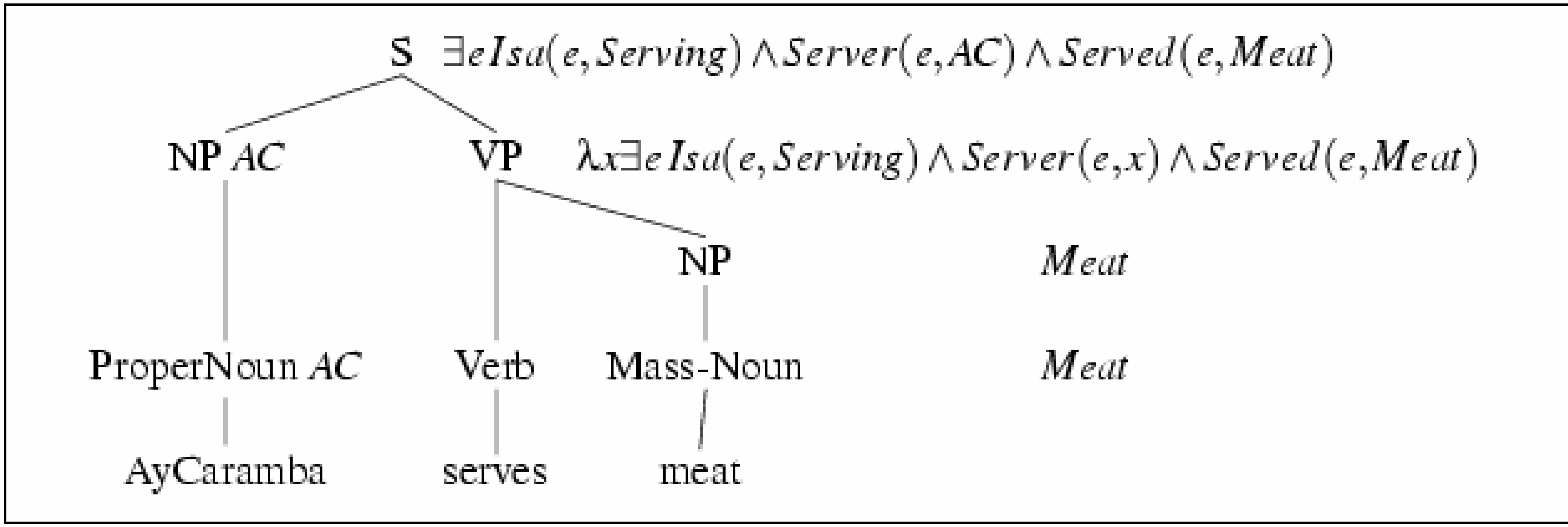
Which FOPC representation is better?

$$\lambda x \lambda y \exists e Serving(e) \wedge Server(e,y) \wedge Served(e,x)$$

$$\lambda x \lambda y \exists e Isa(e, Serving) \wedge Server(e,y) \wedge Served(e,x)$$

Possible pop-quiz: Redo previous example using second representation

Syntax-Driven Semantic Analysis
Semantic Augmentation to CFG Rules



- Revise *Verb* attachment

$$Verb \rightarrow serves\ \{\lambda x\, \lambda y\, \exists e\ Isa(e, Serving) \wedge Server(e, y) \wedge Served(e, x)\}$$

Predicate-Argument Semantics

- The functions/operations permitted in the semantic rules fall into two classes
  - Pass the semantics of a daughter up unchanged to the mother
  - Apply (as a function) the semantics of one of the daughters of a node to the semantics of the other daughters

Predicate-Argument Semantics

- S → NP VP
- VP → Verb NP

- {VP.sem (NP.sem)}
- {Verb.sem (NP.sem)

- in each rule there’s a daughter whose semantics is a function and one that isn’t.

Integration with a Parser

- Assume you're using a dynamic-programming style parser (Earley or CYK).
- As constituents are completed and entered into the table, we compute their semantics.
  - If they're complete, we have their parts.
  - If we have their parts we have the semantics for the parts…
  - Therefore we can compute the semantics of the newly completed constituent.

Mismatches

- There are unfortunately some annoying mismatches between the syntax of FOPC and the syntax provided by our grammars…
- So we'll accept that we can't always directly create valid logical forms in a strictly compositional way

Quantified Phrases

- Consider

  A restaurant serves meat.

- Assume that A restaurant looks like

$$\exists x\ Isa(x, Restaurant)$$

- If we do the normal lambda thing we get

$$\exists e Serving(e) \wedge Server(e, \exists x Isa(x, Restaurant)) \wedge Served(e, Meat)$$

Semantic Augmentation to CFG Rules

- A restaurant serves meat.
  - Subject

    $\exists x\ Isa(x, Restaurant)$
  - Embed in the *Server* predicate:

$$\exists e\ Isa(e, Serving) \wedge Server(e, \exists x\ \boldsymbol{Isa(x, Restaurant)}) \wedge Served(e, Meat)$$

Not a valid FOPC

Semantic Augmentation to CFG Rules

- Solve this problem by introducing the notion of a **complex-term**.
  - A complex term: $< \textit{Quantifier variable body} >$

$$\exists e\ Isa(e, Serving) \wedge Server(e, \boldsymbol{<\exists x\ Isa(x, Restaurant>})) \wedge Served(e, Meat)$$

  - Rewriting a predicate using a complex-term

$$P(< \textit{Quantifier variable body} >) \Rightarrow$$
$$\textit{Quantifier variable body Connective } P(\textit{variable})$$

$$Server(e, < \exists x\ Isa(x, Restaurant>) \Rightarrow \exists x\ Isa(x, Restaurant) \wedge Server(e, x)$$

Complex Terms

- Allow the compositional system to pass around representations like the following as objects with parts:

$$< \exists x \; Isa(x, \text{Restaurant}) >$$

`Complex-Term : <Quantifier var body>`

Example

- Our restaurant example winds up looking like

$$\exists e Serving(e) \wedge Server(e, < \exists x Isa(x, Restaurant) >) \wedge Served(e, Meat)$$

Conversion

- So… complex terms wind up being embedded inside predicates. So pull them out and redistribute the parts in the right way…

P(<quantifier, var, body>)

turns into

Quantifier var body connective P(var)

Example

$$Server(e, <\exists x\ Isa(x, \text{Restaurant})>)$$

$$\Rightarrow$$

$$\exists x\ Isa(x, \text{Restaurant}) \wedge Server(e, x)$$

Quantifiers and Connectives

- If the quantifier is an existential, then the connective is an ^ (and)

- If the quantifier is a universal, then the connective is an => (implies)

Multiple Complex Terms

- Note that the conversion technique pulls the quantifiers out to the front of the logical form…
- That leads to ambiguity if there's more than one complex term in a sentence.

Multiple Complex Terms

- Every restaurant has a menu.

$$\exists e\ Isa(e, Having)$$
$$\wedge\ Haver(e, < \forall x\ Isa(x, Restaurant) >)$$
$$\wedge\ Had(e, < \exists y\ Isa(y, Menu) >)$$

$$\forall x\ Restaurant(x) \Rightarrow$$
$$\exists e, y \wedge Isa(e, Having) \wedge Haver(e, x)$$
$$\wedge\ Isa(y, Menu) \wedge Had(e, y)$$

$$\exists y\ Isa(y, Menu) \wedge \forall x\ Isa(x, Restaurant) \Rightarrow$$
$$\exists e\ Isa(e, Having) \wedge Haver(e, x) \wedge Had(e, y)$$

Try to simplify this multiple complex term

Multiple Complex Terms

- The problem of ambiguous **quantifier scoping** — a single logical formula with two complex-terms give rise to two distinct and incompatible FOPC representations.

Ambiguity

- The number of possible interpretations goes up exponentially with the number of complex terms in the sentence
- The best we can do is to come up with weak methods to prefer one interpretation over another

Attachments for a Fragment of English Sentences

- Flight 487 serves lunch.

  $S \rightarrow NP\ VP\ \{DCL(VP.sem(NP.sem))\}$

- Serve lunch.

  $S \rightarrow VP\ \{IMP(VP.sem(DummyYou))$

  $IMP(\exists e Serving(e) \wedge Server(e, DummyYou) \wedge Served(e, Lunch)$

  Imperatives can be viewed as a kind of **speech act**.

For Reference

- Does Flight 207 serve lunch?

  $S \rightarrow Aux\ NP\ VP\ \{YNQ(VP.sem(NP.sem))\}$

  $YNQ(\exists e Serving(e) \wedge Server(e, Flt207) \wedge Served(e, Lunch)$

- Which flights serve lunch?

  $S \rightarrow WhWord\ NP\ VP\ \{WHQ(NP.sem.var, VP.sem(NP.sem))\}$

  $WHQ(x, \exists e, x\ Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Lunch) \wedge Isa(x, Flight))$

Attachments for a Fragment of English Sentences

- How can I go from Minneapolis to Long Beach?

$$S \rightarrow WhWord\ Aux\ NP\ VP\ \{WHQ(WhWord.sem,\ VP.sem(NP.sem))\}$$

$$WHQ(How, \exists e\ Isa(e, Going) \wedge Goer(e, User) \wedge Origin(e, Minn) \wedge Destination(e, LongBeach))$$

Attachments for a Fragment of English
NPs: Compound Nominals

- The meaning representations for NPs can be either normal FOPC terms or complex-terms.
- Flight schedule
- Summer flight schedule

$$Nominal \rightarrow Noun$$

$$Nominal \rightarrow Nominal\ Noun\ \{\lambda x\ Nominal.sem(x) \wedge NN(Noun.sem, x)\}$$

$$\lambda x\ Isa(x, Schedule) \wedge NN(x, Flight)$$

$$\lambda x\ Isa(x, Schedule) \wedge NN(x, Flight) \wedge NN(x, Summer$$

Attachments for a Fragment of English
NP: Genitive NPs

- (Ex.) Atlanta's airport
- (Ex.) Maharani's menu

$$NP \rightarrow ComplexDet\ Nominal$$

$$\{<\exists x Nominal.sem(x) \wedge GN(x, ComplexDet.sem)>\}$$

$$ComplexDet \rightarrow NP's\ \{NP.sem\}$$

$$<\exists x\ Isa(x, Airport) \wedge GN(x, Atlanta)>$$

Attachments for a Fragment of English
Adjective Phrases

- I don't mind a cheap restaurant.
- This restaurant is cheap.
- For pre-nominal case, an obvious and *often incorrect* proposal is:

$Nominal \rightarrow Adj\ Nominal$

$\{\lambda x\ Nominal.sem(x) \wedge Isa(x, Adj.sem)\}$

$Adj \rightarrow cheap\ \{Cheap\}$

$\lambda x\ Isa(x, Restaurant) \wedge Isa(x, Cheap)$ **intersective semantics**

- ***Wrong***
  - $small\ elephant \Rightarrow \lambda x\ Isa(x, Elephant) \wedge Isa(x, Small)$
  - $former\ friend \Rightarrow \lambda x\ Isa(x, Friend) \wedge Isa(x, Former)$
  - $fake\ gun \Rightarrow \lambda x\ Isa(x, Gun) \wedge Isa(x, Fake)$

} ***Incorrect interactions***

Attachments for a Fragment of English
Adjective Phrases

- The best approach is to simply note the status of a special kind of modification relation and
  - Assume that some further procedure with access to additional relevant knowledge can replace this vague relation with an appropriate representation.

$$Nominal \rightarrow Adj\ Nominal$$

$$\{\lambda x\ Nominal.sem(x) \wedge AM(x,\ Adj.sem)\}$$

$$Adj \rightarrow cheap\ \{Cheap\}$$

$$\lambda x\ Isa(x,\ Restaurant) \wedge AM(x,\ Cheap)$$

Attachments for a Fragment of English
VPs: Infinite VPs

- In general, the λ-expression attached to the verb is simply applied to the semantic attachments of the verb's arguments.
- However, some special cases, for example, infinite VP
- (15.13) I told Harry to go to Maharani.
  - The meaning representation could be:

$$\exists e, f, x\ Isa(e, Telling) \wedge Isa(f, Going) \wedge Teller(e, Speaker) \wedge Tellee(e, Harry) \wedge ToldThing(e, f) \wedge Goer(f, Harry) \wedge Destination(f, x)$$

  - Two things to note:
    - It consists of two events, and
    - One of the participants, *Harry*, plays a role in both of the two events.

Attachments for a Fragment of English
VPs: Infinite VPs

- A way to represent the semantic attachment of the verb, *tell*

  $\lambda x, y \lambda z \exists e\ Isa(e, Telling) \wedge Teller(e, z) \wedge Tell(e, x) \wedge ToldThing(e, y)$

  - Providing three semantic roles:
    - a person doing the telling,
    - a recipient of the telling, and
    - the proposition being conveyed
  - **Problem:**
    - Harry is not available when the *Going* event is created within the infinite verb phrase.

Attachments for a Fragment of English
VPs: Infinite VPs

- Solution:

$$VP \rightarrow Verb \;\; NP \;\; VPto \quad \{Verb.sem(NP.sem,\; VPto.sem)\}$$

$$VPto \rightarrow to \; VP \; Verb \;\; NP \; \{VP.sem\}$$

$$Verb \rightarrow tell \quad \{\lambda x, y \, \lambda z \; \exists e, y.variable \; Isa(e, Telling) \wedge Teller(e, z) \wedge Tellee(e,x) \wedge ToldThing(e, y.variable) \wedge y(x)\}$$

For Reference

  - The λ-variable *x* plays the role of the *Tellee* of the telling and the argument to the semantics of the infinitive, which is now contained as a λ-expression in the variable *y*.
  - The expression *y*(*x*) represents a λ-reduction that inserts *Harry* into the *Going* event as the *Goer.*
  - The notation *y.variable* is analogous to the notation used for complex-terms variables, and gives us access to the event variable representing *Going* event within the infinitive's meaning representation.

Attachments for a Fragment of English

Prepositional Phrases

- At an abstract level, PPs serve two functions:
  - They assert binary relations between their heads and the constituents to which they attached, and
  - They signal arguments to constituents that have an argument structure.
- We will consider three places in the grammar where PP serve these roles:
  - Modifiers of NPs
  - Modifiers of VPs, and
  - Arguments to VPs

Attachments for a Fragment of English
PP: Nominal Modifier

- (15.14) A restaurant on Pearl

$$\exists x\ Isa(x, Restaurant) \wedge On(x, Pearl)$$

$$NP \rightarrow Det\ Nominal$$

$$Nominal \rightarrow Nominal\ PP \quad \{\lambda z\ Nominal.sem(z) \wedge PP.sem(z)\}$$

$$PP \rightarrow P\ NP \quad \{P.sem(NP.sem)\}$$

$$P \rightarrow on \quad \{\lambda y\, \lambda x\ On(x,y)\}$$

Attachments for a Fragment of English
PP: VP Modifier

- (Ex.) ate dinner in a hurry

  $VP \rightarrow VP\ PP$

  - The meaning representation of *ate dinner*

    $\lambda x \exists e\ Isa(e, Eating) \wedge Eater(e, x) \wedge Eaten(e, Dinner)$
  - The representation of the PP

    $\lambda x\ In(x, <\exists h\ Hurry(h)>)$
  - The correct representation of the modified VP should contain the conjunction of the two
    - With the *Eating* event variable filling the first argument slot of the *In* expression.

    $VP \rightarrow VP\ PP\ \{\lambda y\ VP.sem(y) \wedge PP.sem(VP.sem.variable)\}$
  - The result of application

    $\lambda y \exists e\ Isa(e, Eating) \wedge Eater(e, y) \wedge Eaten(e, Dinner) \wedge In(e, <\exists h\ Hurry(h)>)$

Non-Compositionality

- Unfortunately, there are lots of examples where the meaning (loosely defined) can’t be derived from the meanings of the parts
  - Idioms الأمثال
  - Jokes الدعابة
  - Irony التظاهر بالجهل
  - Sarcasm السخرية
  - Metaphor المجاز أو الاستعارة
  - Metonymy الكناية
  - indirect requests الطلب غير المباشر
- Some Examples in Arabic !!

English Idioms

- Kick the bucket, buy the farm, bite the bullet, run the show, bury the hatchet, etc…
- Lots of these… constructions where the meaning of the whole is either
  - Totally unrelated to the meanings of the parts (kick the bucket)
  - Related in some opaque way (run the show)
  - Kick the bucket: To die
  - buy the farm: to be killed
  - bite the bullet: get serious and do what needs to be done even though you don't want to do it
  - run the show: manage the project
  - bury the hatchet: stop arguing or fighting

Problems with Syntactic-Driven Semantics

- Syntactic structures often don't fit semantic structures very well
  - Important semantic elements often distributed very differently in trees for sentences that mean 'the same'

  I like soup. Soup is what I like.

  - Parse trees contain many structural elements not clearly important to making semantic distinctions
  - Syntax driven semantic representations are sometimes odd

Alternatives?

- Semantic Grammars
- Information Extraction Techniques
- Next time

Student Presentation - Start next time

- Tuesday, May 8
  - Saleh Al-Zaid - Language Model Based Arabic Word Segmentation
- Sunday, May 13
  - Al-Elaiwi Moh'd - Diacritization: A Challenge to Arabic Treebank Annotation and Parsing
  - Naif Al-Abdulhay - The Challenge Of Arabic For NLP/MT
  - Abdul Rahman Al Khaldi - Statistical Transliteration for English-Arabic Cross

Student Presentation

- Tuesday, May 15
  - Turki Bakodah - Building A Modern Standard Arabic Corpus
  - Hassan S. Al-Ayesh - Stemming to improve translation lexicon creation form bitexts
  - Saleh Y. Al-Hudail - A Hidden Markov Model –Based POS Tagger for Arabic

Student Presentation

- Sunday May 20
  - Abbas Al-Julaih - An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modeling
  - AbdiRahman Daoud - Online Arabic Handwriting Recognition Using HMM
  - Shaker Al-Anazi - How Do Search Engines Handle Arabic Queries?
- Tuesday, May 22
  - Hussain AL-Ibrahem - Arabic Tokenization, Part-of-Speech Tagging
  - Ahmed Bukhamsin - Hybrid Method for Tagging Arabic Text
  - Al-Ansari, Naser - Light Stemming for Arabic Information Retrieval

Student Presentation Evaluation

- Attendance
- Student evaluation is a must (including Youself)
- On time – no make up – 10%
- Fill after attending the presentation
- Honestly evaluate the presentation

Student Presentation Evaluation

- Improve some needed skills in real life work?
- Helps the instructor
- Fill a grade out of 10 for each item
- Do not fill unattended presentation
- Do evaluate your presentation
- Partial grades to be deducted for unfilled evaluation

Student Presentation Evaluation

Items – Each out of 10

- Introduction
- Clarity
- Knowledge Depth
- Content
- Delivery
- Preparation
- Organization
- Language Usage
- Result & Conclusion
- Question & Answer
- Over all Evaluation

22-2 How To Give Good Presentation

Thank you

- السلام عليكم ورحمة الله