	Language and Computers Topic 2: Searching	Outline	Language and Computers Topic 2: Searching	Searching	Language and Computers Topic 2: Searching
Language and Computers (Ling 384) Topic 2: Searching Detmar Meurers* Dept. of Linguistics, OSU Autumn 2006	Introduction Test Special Special Annactors Coperators Special Annactors Coperators Beachaging the week Coperators Renking of reactor Renking of reactor Syntax of regular expressions Grage: An example for using regular expressions Grage: An example for using the copera and searching them	Introduction Searching in a Library Catalogue Searching the web Advanced searches with regular expressions	Introduction Test Special Special characters Operators Social characters Operators Ranking freums Evaluating search results Advanced searchers With regular expressions Graph An seample for using regular expressions Graph An seample for using the operator and searching them	<ul> <li>A breathtaking number of information resources are available: books, databases, the web, newspapers,</li> <li>To locate relevant information, we need to be able to search these resources, which often are written texts: <ul> <li>Searching in a library catalogue (e.g., using OSCAR)</li> <li>Searching the web (e.g., using Google)</li> <li>Advanced searching in text corpora (e.g., using regular expressions in Opus)</li> </ul> </li> </ul>	Introduction Edu Searching in a Library Catalogue Spacial character Copenses Improve Sanching the web Copenses Improve sanching Readrag of reachs Enduring casesh results Advanced searchers Synta of regulare expension Graph casesholes Terroposa and searching them
* The course was created together with Markus Dickinson and Chris Brew.					
<ul> <li>Searching in speech</li> <li>One might also want to search for speech, e.g., to find a particular sentence spoken in an interview one only has a recording (audio file) of.</li> <li>With current technology, this is only possible if the interview is transcribed, using the IPA or another writing system.</li> <li>It is, however, already possible to <ul> <li>detect the language of a spoken conversation, e.g., when listening in to a telephone conversation</li> <li>detect a new topic being started in a conversation</li> </ul> </li> <li>In the following, we focus on searching in text.</li> </ul>	1/33 Language and Computers Topic 2: Searching Introduction Text Searching in a Library Catalogue Special characters Coverains Searching the web coverains Searching the web Searching the web coverains Searching the web coverains Searching the web Searching the	<ul> <li>Searching in a library catalogue</li> <li>To find articles, books, and other library holdings, a library generally provides a database containing information on its holdings.</li> <li>OSCAR is the database frontend providing access to the library database at OSU.</li> <li>OSCAR makes it possible to search for the occurrence of literal strings occurring in the author, title, keywords, call number, etc. associated with an item held by the library.</li> </ul>	2/33 Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalogue Searching in the web Operators Searching the web Operators Renking of maxim Parking of maxim Searching searching Renking of maxim Searching searching Renking of maxim Searching the web Operators Searching the web Searching th	<ul> <li>Basic searching in OSCAR</li> <li>Literal strings are composed of characters which naturally must be in the same character encoding system (e.g. ASCII, ISO8859-1, UTF-8) as the strings encoded in the database.</li> <li>For literal strings, OSCAR does not distinguish between upper and lower-case letters (i.e. they aren't so literal after all ;-)</li> <li>Adjacent words are searched as a phrase. <ul> <li>art therapy</li> <li>vitamin c</li> </ul> </li> <li>In addition to querying literal strings, the query language of OSCAR also supports the use of <ul> <li>special characters to abbreviate multiple options</li> <li>special operators for combining two query strings (boolean operators) or modifying the meaning of a single string (unary operators)</li> </ul> </li> </ul>	2/33 Language and Computers Topic 2: Searching Introduction Test Speech Searching in a Library Catalogue Searching the web Operators Correc
OSCAR: Special characters	4/33 Language and Computers Topic 2: Searching	OSCAR: Literal Strings and Operators (I)	5/33 Language and Computers Topic 2: Searching	OSCAR: Operators (II)	6/33 Language and Computers Topic 2: Searching
<ul> <li>Use * for 1–5 characters at end or within a word.</li> <li>art* finds arts, artists, artistic</li> <li>gentle*n</li> <li>Use ** for any number of characters at end of word. art** finds artificial, artillery</li> <li>Use ? for a single character at end or within a word. gentlem?n</li> <li>The special * and ? characters must have at least 2 characters to their left. (→ for efficiency reasons)</li> </ul>	Introduction Test Speech Speech Corretors Speech Corretors Searching the operation Speech and speech Corretors Beaching the operation Ranking of realist Evaluation search results Advanced Speech Speech Syntax of regular expressions Greg. An example for using results of the operation Syntax of regular expressions Greg. An example for using results of the operation of the operation of the operation of the operation of the operation of the operation of the operation of the operati	<ul> <li>Use and or or to specify multiple words in any field, any order.</li> <li>art and therapy</li> <li>art or therapy</li> <li>Use and not to exclude words.</li> <li>art and not therapy</li> </ul>	Introduction Teal Speach Speach Speach Speach Speach Correlation Speach	<ul> <li>Use parentheses to group words together when using more than one operator. art therapy and not ((music or dance) therapy)</li> <li>Use near to specify words within 10 words of each other, in any order.</li> <li>art near therapy</li> <li>Use within n to specify words within n words of each other. The value of n has no limit.</li> <li>art within 12 therapy</li> </ul>	Introduction Text Speech Speech Speech Common Speech Common Searching the web Common Searching t
	7/33		8/33		9/33

Searching the web	Language and Computers Topic 2: Searching	The nature of the web	Language and Computers Topic 2: Searching	Search engines	Language and Computers Topic 2: Searching
<ul> <li>A computer user</li> <li>wants to find something on "the web", i.e., in files accessible via the hypertext transfer protocol (http) protocol on the internet</li> <li>goes to a search engine = program that matches documents to a user's search requests</li> <li>enters a query = request for information</li> <li>gets a list of websites that might be relevant to the query</li> <li>evaluates the results: either picks a website with the information looked for or reformulates the query</li> </ul>	Introduction Text Speech Speech Speak Handler Speak Handler Speak Handler Corrator Banking dreak Hanking dreak Han	<ul> <li>Web pages are generally less structured than a record in a library database (with title, author, subject, and other fields).</li> <li>One generally searches for words found anywhere in the document.</li> <li>It is, however, possible to include meta data in a web page.</li> <li>Meta data is additional, structured information that is not shown in the web page itself: e.g., the language a web page is in, its character encoding, author, keywords, etc.</li> <li>Example for a meta tag: <meta content="vacation, Greece" lang="en-us" name="keywords"/></li> </ul>	Introduction Test Speach Speac	<ul> <li>Search engines (e.g., Google)         <ul> <li>store a copy of all web pages</li> <li>create an index to provide efficient access to this large number of pages (e.g., Google currently searches over 4 billion pages)</li> <li>compute a rank for each web page to be able to rank the query results</li> </ul> </li> <li>Search engines differ in various ways:         <ul> <li>stemming: treat <i>bird</i> and <i>birds</i> as the same or not</li> <li>capitalization: treat <i>trip</i> and <i>Trip</i> the same or not</li> <li>use of operators</li> <li>special interface for advanced searching</li> <li>how search results are ranked</li> <li>clustering: group similar results or not</li> </ul> </li> </ul>	Introduction Test Speech Speec
Google: Operators (I)	10/33 Language and Computers Topic 2: Searching	Google: Operators (II)	11/33 Language and Computers Topic 2: Searching	Google: Advanced searching	12/33 Language and Computers Topic 2: Searching
<ul> <li>+: Require a word to occur in the result</li> <li>e.g., To find a restaurant that serves both tofu and BBQ one could try <ul> <li>+tofu +BBQ</li> </ul> </li> <li>-: Disallow a word from occurring in the result</li> <li>e.g., As a <i>potatos</i> purist :-), I search for <ul> <li>potatos -potatoes</li> </ul> </li> <li>`: Include synonyms of the word</li> </ul> <li>Quotation Marks (phrases) <ul> <li>e.g., looking for sites on <i>What Cheer, lowa</i> with</li> <li>"What Cheer"</li> </ul> </li>	Introduction Text Speech Searching in a Library Catalogue Spead characters Operators Bearching the web Description Reading carding Reading Carding Rea	<ul> <li>intitle: Find words used in a title <ul> <li>e.g., intitle:Buckeye finds only web pages which has this word in the title</li> </ul> </li> <li>inurl: Find words used in the url <ul> <li>e.g., inurl:ling returns more linguistics webpages than ling does</li> </ul> </li> <li>link: Find pages that link to a certain page <ul> <li>e.g., link:www.osu.edu to show pages linking to the main osu web page</li> </ul> </li> <li>site: Find pages that are part of a single domain <ul> <li>e.g., I want to find strange attractions involving fish. Knowing one site which has such stuff, one can try fish site:www.roadsideamerica.com.</li> </ul> </li> </ul>	Introduction Text Speech Speech Speech Childray Catalogue Speed characters Coperators Searching the web Description Realing earching Realing earching Real	<ul> <li>More elaborate web forms are provided as alternative to using operators:</li> <li>match all: matches all terms in your query</li> <li>match any: matches as many terms in your query as it can find <ul> <li>e.g., I'm looking for a restaurant that has <i>bbq</i> or <i>bb-que</i> or <i>barbeque</i> in the title</li> <li>most search engines return "match all" followed by "match any" results</li> <li>exclude: eliminate documents which contain certain words</li> </ul> </li> </ul>	Introduction Text Speech Searching in a Library Catalogue Speators Coa
<ul> <li>Improving searching (I)</li> <li>How can I make my searches better?</li> <li>Be on the watch for ambiguity = one word has multiple meanings <ul> <li>e.g., bed: flower bed, sleeping bed, truck bed</li> </ul> </li> <li>Use synonyms and other related words <ul> <li>e.g., plant: building, complex, works, power (distinguish from flora)</li> </ul> </li> <li>Be aware of stop words = words that some search engines ignore because they are "uninformative," such as <i>the</i>, of, and so on</li> </ul>	13/33	<ul> <li>Improving searches (II)</li> <li>Exclude problematic words <ul> <li>e.g., "jefferson airplane -starship" (if you don't want info on the Starship years)</li> </ul> </li> <li>Be aware of parts of speech and what other guises they come in. <ul> <li>e.g., plant: planting, planter, planted (distinguish from power plant)</li> </ul> </li> <li>Continually narrow your focus (using the feedback) <ul> <li>e.g., Want to find information on the game Hearts</li> <li>hearts: too vague, too many non-card game sites → add a related word</li> <li>hearts cards: better, but still greeting cards listed → I see trick listed on one site's description and realize this makes for a good keyword</li> <li>hearts cards trick: good, but now we get card tricks → time for boolean expressions</li> </ul></li></ul>	14/33  Language and Computers  Topic 2: Searching  Introduction Teil Searching in a Library Catalogue Genetion  Searching in a Library Catalogue Genetion  Bearching in a Library Catalogue Genetion  Computer in the searching  Autopart of	<ul> <li>Ranking of results</li> <li>Ideally, the webpages matching a query are returned as an ordered list based on a page's relevance.</li> <li>How can a search engine, which does not understand language, determine the relevance of a particular page?</li> </ul>	Israe Computers Computers Computers Computers Introduction That Search Computers Compu
	16/33	L	17/33	<u> </u>	18/33

<ul> <li>Information used to rank results</li> <li>Counting the number of links to and from a page, to determine how popular a page is. (As a result, unpopular or new pages require a more specific query to be found.)</li> <li>Keeping track of the nature of links to a page; linked pages might be thematically related.</li> <li>e.g., Even if I never mention Sinclair Lewis on a page describing his book <i>Babbit</i>, it can be identified if many Sinclair Lewis sites link to my page.</li> <li>bonuses/penalties for sites known to be of high/low quality</li> <li>looking for keywords in metadata</li> <li>counting how often a web result was clicked on by a user (click-through measurement)</li> <li>various secret ingredients</li> </ul>	Language and Computers Topic 2: Searching Introduction Weight of the Computer Searching in a Computer Searching the weight Market of the Computer Market of the Computer Market of the Computer Searching for additional Market of the Computer Searching for additional Market of the Computer Market of	<ul> <li>Evaluating search results</li> <li>What measures can one use to evaluate how successful a query is?</li> <li>precision: How many of the pages returned are the ones we want? <ul> <li>e.g., Google gives me 400 hits for a query, 200 of which are related to the topic I want; precision = 50%.</li> <li>recall: How many pages on the topic we wanted were actually given? (hard to calculate for web searching)</li> <li>e.g., Google gave me 200 pages I wanted, but there were actually 1000 pages on that topic out there somewhere on the internet; recall = 20%.</li> </ul> </li> <li>We saw earlier how to use our initial results to refine our query and improve precision</li> </ul>	Language and Computers Topic 2: Searching Introduction Teat Searching in a Library Catalogue Becal dwarders Coreation Becal dwarders Coreation Cor	<ul> <li>Motivating regular expressions</li> <li>If one wants to be able to describe more complex patterns of words and text, sometimes boolean expressions aren't enough:</li> <li>In a large document I want to find addresses with a zip code starting with 911 (around Pasadena, CA); but clearly we would not want to report back all occurrences of emergency phone numbers in the document.</li> <li>I want to find all osu email addresses which occur in a long text.</li> <li>I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5: for Jermaine, I want to accept <i>Germaine, Germane, Jermain,</i> and so on.</li> <li>⇒ It would be nice to have a compact way of representing all of these options.</li> <li>Anything where you have to match a complex pattern so-called <b>regular expressions</b> are useful.</li> </ul>	Language and Computers Totic 2: Searching Introduction Weight of the search Searching in a Library Catalogue Searching the weight Portants Broken generating Researching de search Searching and searching Portant of mydel separation Character Search Searching and searching Portant of mydel separation Capacity of search Searching and searching Portant of mydel separation Capacity of search Search Search Search Search Search Search Search Search Search Search Search Search Search Search Sear
<ul> <li>Regular expressions: What they are</li> <li>A regular expression is a compact description of a set of strings, i.e., a language (in formal language theory).</li> <li>They can be used to search for occurrences of these strings</li> <li>Regular expressions can only describe so-called regular languages.</li> <li>This means that some patterns cannot be specified using regular expressions, e.g., finding a string containing matching left and right parentheses.</li> <li>Note that just like any other formalism, regular expressions as such have no linguistic contents, but they can be used to refer to strings encoding a natural language text.</li> </ul>	<section-header><section-header><section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	<ul> <li>Regular expressions: Tools that use them</li> <li>A variety of unix tools (grep, sed,), editors (emacs, jEdit,), and programming languages (perl, python, Java,) incorporate regular expressions.</li> <li>Implementations are very efficient so that large text files can be searched quickly; but not efficient enough for web searching → no web search engine offers them (yet).</li> <li>The various tools and languages differ w.r.t. the exact syntax of the regular expressions they allow.</li> </ul>	Language and Computers To 2: Searching Beach Beach Beach Beach Commis	<pre>The syntax of regular expressions (I) Regular expressions consist of     strings of literal characters: c, A100, natural     language, 30 years!     disjunction:         ordinary disjunction: devoured ate, famil(y ies)         character classes: [Tt]he, bec[oa]me         ranges: [A-Z] (any capital letter)     negation:     [^a] (any symbol but a)     [^A-Z0-9] (not an uppercase letter or number)</pre>	<section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header>
<ul> <li>The syntax of regular expressions (II)</li> <li>counters <ul> <li>optionality: ? <ul> <li>colou?r</li> <li>any number of occurrences: * (Kleene star)</li> <li>[0-9]* years</li> <li>at least one occurrence: + <ul> <li>[0-9]+ dollars</li> </ul> </li> <li>wildcard for any character: . <ul> <li>beg.n for any character in between beg and n</li> </ul> </li> </ul></li></ul></li></ul>	Language and Computers Topic 2: Searching Introduction Test Speech Searching in a Library Catalogue Spoal characters Coreators Searching the web Operators Searching the web Operators Manage of results Evaluating searching Ranking of results Advanced searches with regular expressions Enter freque results Care, An example units Care, An example units Test corpora and searching them	<ul> <li>The syntax of regular expressions (III)</li> <li>Escaped characters: to specify a character with a special meaning (*, +, ?, (, ),  , [, ]) it is preceded by a backslash (\)</li> <li>e.g., a period is expressed as \.</li> <li>Operator precedence, from highest to lowest: <ul> <li>parentheses ()</li> <li>counters * + ?</li> <li>character sequences</li> <li>disjunction  </li> </ul> </li> </ul>	Language and Computers Topic 2: Searching Introduction Teat Speech Searching in a Library Catalogue Special varacters Operators Bearching the web Operators Bearching the web Operators Bearching the web Constitution Ranking of results Revealing searching Ranking of results Advanced searches with regular expressions System diright separation Computer Searching System Computer Searches With regular System Computer Searches With regular System Computer Searches With regular System Computer Searches With regular System Computer Searches System Computer Searches System Computer Searches The Computer Searches Searching Searching Searches Searching Searching Searches Searching Searches Sea	<ul> <li>Grep</li> <li>grep is a powerful and efficient program for searching in text files using regular expressions.</li> <li>It is standard on Unix, Linux, and Mac OSX, and there also are various ports to Windows (e.g., http://gnuwin32.sourceforge.net/packages/grep.htm, http://www.interlog.com/?tcharron/grep.html or http://www.wingrep.com/).</li> <li>The version of grep that supports the full set of operators mentioned above is generally called egrep (for extended grep).</li> </ul>	Language and Computers Topic 2: Searching Introduction Test Speech Searching in a Library Catalogue Spoat Amartes Operators Impoving searching Ranking of multis Searching the web Operators Impoving searching Ranking of multis Advanced Searches with regular expressions Systa of regular expressions Charge o

<pre>Grep: Examples for using regular expressions (I) In the following, we assume a text file f.txt containing, among others, the strings that we mention as matching.     Strings of literal characters:     egrep 'and' f.txt matches and, Ayn Rand, Candy     and so on     Character classes:     egrep 'the year [0-9][0-9][0-9][0-9]' f.txt     matches the year 1776, the year 1812, the year     2001, and so on     Escaped characters:     egrep 'why\?' f.txt matches why?, whereas     egrep 'why?' f.txt matches why and wh</pre>	Language and Computers Toto Speech Speech Speech Speech Speech Searching in a Library Catalogue Speach churates Operators Improving searching Ranking of results Coreators Searching the web Operators Improving searching Ranking of results Coreators Searching the web Coreators Improving searching Ranking of results Coreators Speach regular expressions Systax of regular expressions (Search expressions) Systax of regular expressions (Search expressions) Systax of regular expressions (Search expressions) Stat of regular expressions) Systax of regular expressions (Search expressions) Systax of regular expressions) Tot dorperate and searching them	<pre>Grep: Examples for using regular expressions (II)</pre>	Language and Computers Tote 2: Searching Tote 2: Searching Tote 3: Searching in a Library Catalogue Special churates Operates Inspeving carding Ranking of results Coreates Searching the web Operates Inspeving carding Ranking of results Cardina of regular expressions Cardina of r	<ul> <li>Grep: Examples for using regular expressions (III)</li> <li>Kleene star (*):     <pre>egrep 'a*rgh' f.txt matches argh, aargh, aaargh     egrep 'sha(la)*' f.txt matches sha, shala,     shalala, or if you're Van Morrison     shalalalalalalala</pre> </li> <li>One or more (+):     <pre>egrep 'john+y' f.txt matches johny, johnny,,     but not johy </pre></li> <li>Optionality (?):     <pre>egrep 'joh?n' f.txt matches jon and john</pre></li> </ul>	Language and Computers Topic 2: Searching Topic 2: Searching Teat Speach Searching in a Library Catalogue Speato Upparton Searching the web Operaton Impoving searching Parking of result Reading searching Parking of result Advanced Searchess with regular expressions Bytace of regular expressions Charge and regular build Charge and searches with regular searchessearches with regular Searchessearches Marchessearches Charge and searches Charge and searches Searching the searches Searchessearches Charge and searches Searchessearchessearches Searchessearchessearches Searchessearchessearches Searchessearchessearchessearches Searchessearchessearchessearchessearches Searchessearche
Corpora	28/33 Language and Computers Topic 2: Searching	How corpora can be searched	29/33 Language and Computers Topic 2: Searching	Exploring regular expressions	30/33 Language and Computers Topic 2: Searching
<ul> <li>A corpus is a collection of text.</li> <li>Corpora with the works of various writers, newspaper texts, etc. have been collected and electronically encoded.</li> <li>Corpora can be quite large</li> <li>The British National Corpus is a 100 million word collection representing a wide cross-section of current written and spoken British English.</li> <li>Another example is the European Parliament Proceedings Parallel Corpus 1996–2003.</li> </ul>	Introduction Texe Beensh Searching in a Library Catalogue Joreans Control Con	<ul> <li>Both the BNC and the European Parliament corpus can be searched using on-line web-forms.</li> <li>Both of the web forms allow regular expressions for advanced searching.</li> <li>To provide efficient searching in large corpora, in these search engines regular expressions over characters are limited to single tokens (i.e. generally words).</li> <li>BNC: <ul> <li>web form: http://sara.natcorp.ox.ac.uk/lookup.html</li> <li>regular expressions are enclosed in { }</li> </ul> </li> <li>European Parliament Corpus: <ul> <li>web form: http://logos.uio.no/cgi-bin/opus/opuscqp.pl? corpus=EUROPARL;lang=en</li> <li>in the simplest case, regular expressions are encosed in " "</li> </ul> </li> </ul>	Introduction Text Text Searching in a Lorange and abaardees Control C	To explore the use of regular expressions, check out http://www.lexmasterclass.com/exercises/regex/index.html which offers exercises with immediate feedback (by showing the matched characters in red).	Introduction Teat Been: Sacchaptang Corrent Sacchaptang Corrent Corren