	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 Winter 2005	Introduction Text Speach Searching in a Library Catalogue Speater Speater Speater Departure Internet Coperators Departure Internet Mankag dreams Evaluating saarch results Advanced searches with regular expressions Sprate dreams Speater Sprate dreams Sprate dreams Status of regular expressions Sprate dreams Status of regular Sprate dreams Sprate dreams Status of regular Sprate dreams Sprate d	Introduction Searching in a Library Catalogue Searching the web Advanced searches with regular expressions	Introduction Tea Speech Searching in a Library Catalogue Special Anautos Operator Portune Improving searching Ranking dreams Evaluating search results Advanced SearChes With regular expressions Operator Ranking and searching Ranking and searching Ranking dreams Advanced Searches With regular expressions Operator Payta of ngular expressions Gez, no example being Teat corpore and searching them	 A breathtaking number of information resources are available: books, databases, the web, newspapers, To locate relevant information, we need to be able to search these resources, which often are written texts: Searching in a library catalogue (e.g., using OSCAR) Searching the web (e.g., using Google) Advanced searching in text corpora (using regular expressions) (e.g., using Opus) 	Introduction Text Searching in a Library Catalogue Speal character Operators Searching the web Operators Text of the searching Fuelding searching the Searching the web Searching the searching Searching the sea
	1/32		2/32		3/32
 Searching in speech 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. With current technology, this is only possible if the interview is transcribed, using the IPA or another writing system. It is, however, already possible to detect the language of a spoken conversation, e.g., when listening in to a telephone conversation detect a new topic being started in a conversation In the following, we focus on searching in text. 	<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></section-header></section-header>	 Searching in a library catalogue To find articles, books, and other library holdings, a library generally provides a database containing information on its holdings. OSCAR is the database frontend providing access to the library database at OSU. 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. 	<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></section-header></section-header>	 Basic searching in OSCAR 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. For literal strings, OSCAR does not distinguish between upper and lower-case letters (i.e. they aren't so literal after all ;-) Adjacent words are searched as a phrase. art therapy vitamin c In addition to querying literal strings, the query language of OSCAR also supports the use of special characters to abbreviate multiple options special operators for combining two query strings (boolean operators) or modifying the meaning of a single string (unary operators) 	Language and Computers Topic 2: Searching Introduction Iter Participation Computers Co
OSCAR: Special characters	Language and Computers Topic 2: Searching Introduction	OSCAR: Literal Strings and Operators (I)	Language and Computers Topic 2: Searching Introduction	OSCAR: Operators (II)	Language and Computers Topic 2: Searching Introduction
 Use * for 1-5 characters at end or within a word. art* finds arts, artists, artistic gentle*n Use ** for any number of characters at end of word. art** finds artificial, artillery Use ? for a single character at end or within a word. gentlem?n The special * and ? characters must have at least 2 characters to their left. (→ for efficiency reasons) 	Test Speech Searching in a Library Catalogue Operators Searching the web Operators Researching the web Operators Researching the web Researching t	 Use and or or to specify multiple words in any field, any order. art and therapy art or therapy C+ or C++ Use and not to exclude words. art and not therapy 	Test Speech Searching in a Library Catalogue Special drautes Secal drautes Bearching the web Correstors Test and the second Test and the second Searching the web Evaluating search results Advanced Searches With regular expressions Greg. An example for using regular expressions Greg. An example for using regular expressions Test corpora and searching them	 Use parentheses to group words together when using more than one operator. art therapy and not ((music or dance) therapy) Use near to specify words within 10 words of each other, in any order. art near therapy Use within n to specify words within n words of each other. The value of n has no limit. art within 12 therapy 	Ted Speech Searching in a Library Catalogue Speat drawates Department Searching the web Operative Termonia searching Ranking of reals Evaluating search results Advanced searches with regular expressions Organization Organization Syntax of regular expension Organization Syntax of regular expension Organization Charles and the searching Teach of the searching Syntax of regular expension Organization Syntax of regular expension

8/32

9/32

7/32

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
 A computer user wants to find something on "the web", i.e., in files accessible via the hypertext transfer protocol (http) protocol on the internet goes to a search engine = program that matches documents to a user's search requests enters a query = request for information gets a list of websites that might be relevant to the query evaluates the results: either picks a website with the information looked for or reformulates the query 	Introduction Text Speach Speach Annual Annual Speach Annual Speach Barban Barba	 Web pages are generally less structured than a record in a library database (with title, author, subject, and other fields). One generally searches for words found anywhere in the document. It is, however, possible to include meta data in a web page. 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. Example for a meta tag: <meta <br="" name="keywords"/>lang="en-us" content="vacation, Greece"> 	Introduction Text Speach Speach Annual Annual Speach Annual Speach Annual Speach Annual Speach Speach Annual Speach Speac	 Search engines (e.g., Google) store a copy of all web pages create an index to provide efficient access to this large number of pages (e.g., Google currently searches over 4 billion pages) compute a rank for each web page to be able to rank the query results Search engines differ in various ways: stemming: treat <i>bird</i> and <i>birds</i> as the same or not capitalization: treat <i>trip</i> and <i>Trip</i> the same or not use of operators special interface for advanced searching how search results are ranked clustering: group similar results or not 	Introduction Teal Speach Searching in a Library Catalogue Speacid maakers Operation Improvement Improvement Manking of retask Evaluating searching Ranking of retask Evaluating searching Advisories Operation Systa chargen and searching them
Google: Operators (I)	10/32 Language and Computers Topic 2: Searching	Google: Operators (II)	11/32 Language and Computers Topic 2: Searching	Google: Advanced searching	12/32 Language and Computers Topic 2: Searching
 +: Require a word to occur in the result e.g., To find a restaurant that serves both tofu and BBQ one could try +tofu +BBQ. -: Disallow a word from occurring in the result e.g., As a <i>potatos</i> purist, I search for potatos -potatoes `: Include synonyms of the word Quotation Marks (phrases) e.g., "What Cheer" when looking for sites on What Cheer, lowa 	Introduction Test Speech Searching in a Library Catalogue Speators Scatching the web Control International Control Raving of reaching Raving of reaching Raving of reaching Scatching search results Advanced searchess with regular expressions Grap. An example for using regular expressions Tate copone and searching Bern	 intitle: Find words used in a title e.g., intitle:Buckeye finds only web pages which has this word in the title inurl: Find words used in the url e.g., inurl:ling returns more linguistics webpages than ling does link: Find pages that link to a certain page e.g., link:www.osu.edu to show pages linking to the main osu web page site: Find pages that are part of a single domain 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. 	Introduction Text Seech Searching in a Library Catalogue Searching the web Coenter Internet Searching Raving of reaching Raving of reaching Raving of reaching Searching Searching With regular Swith regular Syntax of regular expressions Grage. An example for using regular expressions Taxte copose and searching Ben	 More elaborate web forms are provided as alternative to using operators: match all: matches all terms in your query match any: matches as many terms in your query as it can find e.g., I'm looking for a restaurant that has bbq or bb-que or barbeque in the title most search engines return "match all" followed by "match any" results exclude: eliminate documents which contain certain words 	Introduction Text Searching in a Library Catalogue Social chanacias Operations Scatching the web <u>Desenter</u> Improving searching Ranking of reachs Evaluating search results Advanced Searching With regular expressions Syntax of regular pressions Graph an example for unit regular expressions Text corpor and searching them
 Improving searching (I) How can I make my searches better? Be on the watch for ambiguity = one word has multiple meanings e.g., bed: flower bed, sleeping bed, truck bed Use synonyms and other related words e.g., plant: building, complex, works, power (distinguish from flora) Be aware of stop words = words that search engines ignore because they are "uninformative," such as <i>the</i>, of, and so on e.g., <i>The Police</i> won't help you find the rock band any more than <i>Police</i> will 	<section-header> 13/32 Laguage and Logic 2: Searching Introduction Tage 3: Searching in a Control of the S</section-header>	 Improving searches (II) Exclude problematic words e.g., "jefferson airplane -starship" (if you don't want info on the Starship years) Be aware of parts of speech and what other guises they come in. e.g., plant: planting, planter, planted (distinguish from power plant) Continually narrow your focus (using the feedback) e.g., Want to find information on the game Hearts hearts: too vague, too many non-card game sites → add a related word 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 hearts cards trick: good, but now we get card tricks → time for boolean expressions 	<section-header> 14/32 Laguage and supported Topic 2: Searching Introduction Way Searching in a Lipcary Catalogue Darior Barroy Catalogue Darior Darior</section-header>	 Ranking of results Ideally, the webpages matching a query are returned as an ordered list based on a page's relevance. How can a search engine, which does not understand language, determine the relevance of a particular page? 	15/32 Digit 2: Searching Digit 2: Searching Introduction Big Big Big Big Big Big Big Big
	16/32	1	17/32	1	18/32

 Information used to rank results 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.) Keeping track of the nature of links to a page; linked pages might be thematically related. 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. bonuses/penalties for sites known to be of high/low quality looking for keywords in metadata counting how often a web result was clicked on by a user (click-through measurement) various secret ingredients 	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalogue Speatal Autority Copeators Searching the web Operators Searching the web Operators Relating searching Raining disearching Advanced Gesarches With regular expressions Synta d'explair expressions Synta d'explair expressions Synta d'explair expressions Synta d'explair expressions Synta d'explair expressions Synta d'explair expressions	 Evaluating search results What measures can one use to evaluate how successful a query is? precision: How many of the pages returned are the ones we want? e.g., Google gives me 400 hits for a query, 200 of which are related to the topic I want; precision = 50%. recall: How many pages on the topic we wanted were actually given? (hard to calculate for web searchin) 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%. We saw earlier how to use our initial results to refine our query and improve precision 	Language and Computers Topic 2: Searching Introduction Test Searching in a Library Catalogue Secarching the web Operators Bearching the web Coreators Bearching the web Coreators Coreators Bearching the web Coreators Bearching the web Coreators Bearching the web Coreators State Coreators Bearching the web Coreators Bearching the web Coreators Bearching the web Coreators Bearching the web Coreators Bearching the web Coreators State Coreators Bearching the web Coreators Bearching the web Coreators State Coreators Bearching the web State Coreators Bearching the web St	 Motivating regular expressions If one wants to be able to describe more complex patterns of words and text, sometimes boolean expressions aren't enough: 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. I want to find all osu email addresses which occur in a long text. 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. ⇒ It would be nice to have a compact way of representing all of these options. Anything where you have to match a complex pattern so-called regular expressions are useful. 	Language and Computers Topic 2: Searching Introduction Teat Searching in a Library Catalogue Secarching the web Operates Searching the web Operates Ranking derak Ranking
 Regular expressions: What they are A regular expression is a compact description of a set of strings, i.e., a language (in formal language theory). They can be used to search for occurrences of these strings Regular expressions can only describe so-called regular languages. This means that some patterns cannot be specified using regular expressions, e.g., finding a string containing any number of as followed by exactly the same number of bs. 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. 	<page-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></page-header>	 Regular expressions: Tools that use them A variety of unix tools (grep, sed,), editors (emacs,), and programming languages (perl, python,) incorporate regular expressions. 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). The various tools and languages differ w.r.t. the exact syntax of the regular expressions they allow. 	<page-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></page-header>	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)	24/32
 The syntax of regular expressions (II) counters optionality: ? colou?r any number of occurrences: * (Kleene star) [0-9]* years at least one occurrence: + [0-9]+ dollars wildcard for any character: . beg.n for any character in between beg and n 	Language and Computers Topic 2: Searching Introduction Test Seerch Searching in a Library Catalogue Speatos Library Catalogue Speatos Bearching the web Operatos Evaluating searching Ranking of results Evaluating searching Ranking of results Evaluating searching With regular expressions Sufficience and searching them Part of the provider of the searching them	 The syntax of regular expressions (III) Escaped characters: to specify a character with a special meaning (*, +, ?, (,), , [,]) it is preceded by a backslash (\) e.g., a period is expressed as _ Operator precedence, from highest to lowest: parentheses () counters * + ? character sequences disjunction 	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalogue Speators Improving starting Ranking of results Corestors Searching the web Operators Improving starting Ranking of results Evaluating searching Ranking of results Advanced Searchess with regular expressions Text corpore and searching them	 Grep grep is a powerful and efficient program for searching in text files using regular expressions. It is standard on Unix, Linux, and Mac OSX, and there also are various ports to Windows (e.g., http://gnuvin32.sourceforge.net/packages/grep.htm, http://www.interlog.com/Tcharron/grep.html or http://www.wingrep.com/). The version of grep that supports the full set of operators mentioned above is generally called egrep (for extended grep). 	Language and Computers Topic 2: Searching Introduction Text Speach Searching in a Library Catalogue Speat Untraver Searching the web Operator Searching the web

Grep: Examples for using regular expressions (I)	Language and Computers Topic 2: Searching Introduction	Grep: Examples for using regular expressions (II)	Language and Computers Topic 2: Searching Introduction	Grep: Examples for using regular expressions (III)	Language and Computers Topic 2: Searching Introduction
 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 	Ted Speech Searching in a Library Catalogue Speator Sp	 disjunction (): egrep 'G g' f.txt matches G or g, so egrep 'G gouda' f.txt matches gouda or Gouda. Note that (G g) ouda has the same effect. grouping with parentheses: egrep 'un(interest excit)ing' f.txt matches uninteresting or unexciting. Any character (.): egrep 'o.e' f.txt matches ore, one, ole 	Ted Speech Secarching in a Library Catalogue Special drawates Operators Beaching the web Coperators Inspection Ranking of reactions Ranking of reactions Ran	 Kleene star (*): egrep 'a*rgh' f.txt matches argh, aargh, aaargh egrep 'sha(la)*' f.txt matches sha, shala, shalala, or if you're Van Morrison shalalalalalalalala One or more (+): egrep 'john+y' f.txt matches johny, johnny,, but not johy Optionality (?): egrep 'joh?n' f.txt matches jon and john 	Test Speach Speach Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestins Corestina Coresti
	28/32		29/32		30/32
Corpora	Language and Computers Topic 2: Searching	How corpora can be searched	Language and Computers Topic 2: Searching		
 A corpus is a collection of text. Corpora with the works of various writers, newspaper texts, etc. have been collected and electronically encoded. Corpora can be quite large The British National Corpus is a 100 million word collection representing a wide cross-section of current written and spoken British English. Another example is the European Parliament Proceedings Parallel Corpus 1996–2003. 	Introduction Twe Seesi Sees	 Both the BNC and the European Parliament corpus can be searched using on-line web-forms. Both of the web forms allow regular expressions for advanced searching. To provide efficient searching in large corpora, in these search engines regular expressions over characters are limited to single tokens (i.e. generally words). BNC: web form: http://sara.natcorp.ox.ac.uk/lookup.html regular expressions are enclosed in { } European Parliament Corpus: web form: http://logos.uio.no/cgi-bin/opus/opuscqp.pl? corpus=EUROPARL:lang=en in the simplest case, regular expressions are encosed in " " 	Introduction Twe Been Sector S		