Applying Corpus Linguistics Techniques to Learning Analytics

Russell Entrikin
Brandeis University
russell.entrikin@gmail.com

1 An Introduction to Learning Analytics

1.1 What is Learning Analytics?

One of the difficulties in explaining learning analytics is the elusiveness of a single clear statement of the scope and purpose of the field. A definition was developed in 2011 at the First International Conference on Learning Analytics and Knowledge in Banff, Alberta:

> Learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs.[1]

This is a good start, but reflects the somewhat scattered nature of current work in learning analytics, leaving many questions unanswered. What sorts of data do we need, specifically? How do we go about collecting it? What do we do with it once we have it? Data collection and analysis is nothing new - what does learning analytics give students and instructors that they do not have already?

Perhaps a better strategy is to explain what learning analytics is *not*. Learning analytics developed out of, but is distinct from, fields such as business analytics and academic analytics.[2] Business analytics (also referred to as business intelligence) concerns itself with the analysis of data collected on a large scale, e.g. on both internal and external human resources, finances, etc., to find patterns and correlations which can be exploited to optimize business efficiency and profitability. Academic analytics is essentially business analytics applied to the academic operations of a university, with added dimensions such as student retention, grades, and alumni donations.

Whereas the target audience of business and academic analytics is generally administration or management staff of a given institution, learning analytics is for learners and instructors. Indeed, the fundamental goals of academic analytics and learning analytics are different: while academic analytics focuses on optimizing the academic units of the university as a whole, learning analytics provides feedback on level of the course and the individual student, with the goal of improving learning and staging interventions as needed.

Analysis of student learning is something that universities already have in the form of tests, papers, homework, presentations, course evaluations, teacher feedback, etc. However, graded assignments can only demonstrate that something has gone wrong after the fact, at which time it may be too late to get students back on track. The goal of a learning analytics system is to not only make realtime inferences about the cognitive states of learners - applied mastery vs. rote memorization vs. fundamental lack of understanding - but also to be able to "peek" at future learning outcomes using the current data. This is what learning analytics promises universities that has never been available before: a near-realtime "ticker" of student cognition.

It needs to be emphasized here, to assuage the fears of students and teachers alike, that the goal of learning analytics is not to predict or assign grades. Grades themselves are metrics for student learning - designing one metric to predict another is of limited usefulness. Instead, learning analytics seeks

[1] http://www.learninganalytics.net/?p=126

[2] Analytics in Higher Education: Establishing a Common Language.

to measure learning itself as directly as possible by looking at student behaviors and communication. In fact, automated systems may be able to accurately flag at-risk students faster than even a very viligant professor ever could, particularly in large "gateway" courses. Health insurance companies are already trying to use data-driven approaches to determine who will get sick in the near future:[3] if successful, similar algorithms and types of data can be applied to student data to predict which students will run into trouble, enabling instructors to intervene.

Learning analytics systems are intended to be another "tool" in the toolbox of professors and students, just as quizzes and papers now are. As with all artificial intelligence/analytics systems, there is a limit to what is "knowable" (to an AI or a human) solely by examining metrics. These sorts of systems should be seen as a way of helpfully drawing focus to problematic elements of a university course, and not as a way of automating judgements about students or course material.

Possible applications for learning analytics systems of the future include: systems for instructors which automatically flag student assignments which are likely to have severe problems; systems for students which point out logical and rhetorical flaws in their own work; systems for peer tutors which can anticipate which parts of lessons are difficult and help make targeted recommendations of learning resources; systems for administrators which look beyond grades and evaluate how much actual learning is happening in different courses; and systems for college-bound high school students that can objectively compare student learning between universities (and perhaps plot graphs of learning vs. tuition). A learning analytics-driven approach could detect cheating, not by detecting similarities between the work of other students or Internet text - as current systems do - but rather by flagging work which has a "style" unlike a student's previous work, detecting users of online essays-for-sale schemes[4]. For graduate students, one can imagine a learning analytics system which would suggest journal articles relevant to a given project, which would be extremely helpful for research in less well-traveled areas of study.

1.2 What Sort of Data is Needed for a Robust Learning Analytics System?

"Big data" denotes an overwhelming abundance of available information far exceeding the ability of humans to process, interpret, and sensibly act upon without the use of computers.[5] Universities excel at collecting data about their students, but until now, they have lagged behind the business sector in terms of knowing how to leverage it into optimizing performance (both on the campus and classroom level).[6]

Most of the data necessary for a robust learning analytics system is already available in machine-readable format. In fact, the enormous size of current data sets sometimes makes access and analysis through common database systems like SQL infeasible, and requires systems like Apache Hadoop[7] (which Facebook currently uses)[8] for efficient processing. Lack of data will certainly not be the problem in a real-world implementation of a learning analytics system: if anything, it may be necessary to pare down these massive data sets, choosing only the most informative elements, rather than overwhelming computer systems (and instructors viewing the processed output) with irrelevant information.

Possible informative data resources will vary based on the task of the system (prediction of future student success, day-to-day analysis of performance in a course, broader analysis spanning many years of a student's career, etc.), but could include student grades, demographic information, course syllabi, coursework submitted online, social networking data, and even seemingly tangentially related data on judicial offenses and engagement in campus clubs, which has shown to be one informative predictor of university retention rates.[9]

Selection of data to use in a learning analytics system will likely be a give-and-take process. On the one hand, it is desirable for the sake of computa-

[3] http://www.heritagehealthprize.com/c/hhp

[4] http://www.tailoredessays.com

[5] http://www.wired.com/science/discoveries/magazine/16-07/pb_visualizing

[6] http://www.educause.edu/EDUCAUSE+Review/EDUCAUSE ReviewMagazineVolume46/PenetratingtheFogAnalyticsinLe/235017

[7] http://hadoop.apache.org/

[8] http://www.facebook.com/notes/paul-yang/moving-an-elephant-large-scale-hadoop-data-migration-at-facebook/10150246275318920

[9] NERLA 2012 Proceedings.

tional tractability to use only the most informative data, but on the other hand, the nature of "big data" makes it impossible to know *a priori* which kinds of data will be informative without running them through a computer system. New tools are being developed to address problems like these and efficiently find subtle correlations in large data sets.[10]

1.3 The Social Networking Element: Case Studies

The greatest story in technology in recent years has been the rapid ascent of social networking tools like Twitter and Facebook. Once dismissed as time-wasters, these powerful resources have become an invaluable resource to political and social movements around the world.[11]

While some companies use data mining techniques on social media, universities have been slow to catch on. Some classes have taken to using online forums as a way for students to get in touch with each other and exchange ideas, but information on forum use has largely been piled upon the heap of "big data" and ignored. However, a few groups of trailblazers around the world have more recently developed frameworks that allow rich analysis of connections between students and their ideas.

One exciting project is SNAPP[12] (Social Networks Adapting Pedagogical Practice) at the University of Wollongong in Australia. SNAPP is designed as an interface with existing learning management systems like Moodle and BlackBoard. While current systems can easily give an instructor the "low-hanging fruit" of student metrics (e.g. time spent logged in to the system, number of files accessed, etc.), SNAPP presents graph-based visualizations of forum data, allowing professors to detect at a glance the degree to which students are connected to the network and are getting involved in the discussion. SNAPP is a great example of what is possible without any advanced artificial intelligence or predictive algorithms: the idea is to give instructors an easy-to-interpret visual "dashboard" for analyzing what is going on in classrooms. Tools like these, though not particularly technically impressive or difficult to develop, can be a game-changing resource for teachers and help distill information about complex systems into something more manageable.

"Dashboard" systems for instructors are easy enough to create, but learning analytics will really come into its own once students have an active, collaborative role in the system. Unfortunately, no one seems to know exactly how to get students involved. One attempt is the Check My Activity tool at the University of Maryland[13], which allows students to compare statistics about their online access of course materials to other (anonymous) students, grouped by the grade that they received. The idea behind this is that at-risk students who do not often use the learning management system will check their activity online, realize that top-tier students access the system much more frequently, and seek to change their ways. Several criticisms could be leveled at such an approach, among them, that students who do not often use the online system will be unlikely to check their activity in any case, and that analyzing online activities in terms of mere usage time is reductive. However, Check My Activity makes the important first step of providing students a way of examining and reflecting upon their own behavior. One of the goals of learning analytics is to start a larger dialogue about how students learn: Check My Activity has a chance at achieving this, especially for inexperienced freshmen in large classes.

A more versatile and open-ended system is LOCO-Analyst[14], which provides in-depth metrics for students' online activities. Like SNAPP, the key here is readable, customizable visualizations, allowing students and teachers unprecendented access to statistics about their own activities in a way that makes sense to them. The system not only provides detailed metrics per student, per quiz/test, and per learning module, but also allows students to collaboratively "tag" modules with important concepts. LOCO-Analyst still seems primarily targeted towards instructors, but features like student collaborative tagging are the beginnings of a transition from a mere monitoring device to a next-generation

[10] http://www.sciencemag.org/content/334/6062/1518

[11] http://fletcher.archive.tusm-oit.org/forum/archives/pdfs/35-2pdfs/Dunn_FA.pdf

[12] http://research.uow.edu.au/learningnetworks/seeing/snapp/index.html

[13] http://www.educause.edu/EDUCAUSE+Quarterly/EDUCAUSEQuarterlyMagazineVolum/VideoDemoofUMBCsCheckMyActivit/219113

[14] http://jelenajovanovic.net/LOCO-Analyst/

learning analytics system where students work together to learn. One known issue in this sort of a system is balancing detail with readability: some instructors will invariably be intimidated by the vast amount of charts and tables available.[15] A possible solution is to make systems highly customizable, allowing each instructor to hide metrics they do not find informative. Denser, possibly less user-friendly systems like LOCO-Analyst have a lot to offer instructors, but will also likely require lessons for teachers and students on how to use them to their fullest potential.

The most ambitious and most promising learning analytics system in development now is the Cohere web application.[16] The idea behind Cohere is that linguistic theory about discourse modeling can be applied to represent learning as a series of "rhetorical moves" (e.g. disagreeing, presenting a new idea, providing more evidence, etc.).[17] Users create a network of concepts, annotating connections with rhetorical move labels as they go (thereby reflecting on their own argumentation style) and collaboratively evolving a framework of "collective thought". Students can link directly to web sources which they find useful. Instead of posting their thoughts in one-dimensional chronological order, as on Facebook/Twitter/online forums, collaborative learning is represented as a two-dimensional graph showing the structure of discourse, with ideas as nodes and rhetorical moves as links. Concept networks (between ideas) and social networks (between students) are built up in parallel, adding a third dimension, and other similar projects even add a temporal dimension, allowing a "playback" of the evolution of the network over time.[18] Eventually, it may even be possible to use this kind of data to bootstrap a learning analytics system into automatically labeling existing documents with rhetorical moves and representing them as explorable networks of ideas.[19]

A possible criticism of Cohere is that it is perhaps too open-ended. Creation of concept networks is highly customizable, but without a more rigid structure, it may be difficult to make analysis of such networks computational. We will return to issues like this one when discussing the annotation process in corpus linguistics.

1.4 Predictive Systems and Artificial Intelligence

There are fewer existing examples of predictive learning analytics systems, but one attempt is Purdue's Course Signals.[20] Course Signals uses data from Blackboard - grades in the class so far, time spent using online resources, and students' past academic performance - to alert students who are underperforming. This is a clever idea, but the extent to which such a system can provide helpful, concrete feedback to students is unclear. The user-friendly presentation of the results of analysis as a "traffic light" graphic - with red, yellow, and green corresponding to different alert levels - is a bit simplistic, and although clicking the traffic light displays suggestions about how to modify study habits, it is dubious how targeted and constructive these suggestions could possibly be when based only on data drawn from Blackboard.

This raises the issue of auditability in predictive/artificial intelligence systems. A computer system which predicts weather only needs to predict when a storm will hit - knowing which meteorological factors contributed to the storm is a nice bonus for scientists, but the main purpose of the system is to warn people about an inevitable event so that they can make preparations. This is not the case in learning analytics, where decision-making computers are only useful when they are able to explain why a student, assignment, or learning module was assigned a certain label, i.e. when their decisions are auditable. There are two main reasons for this: firstly, even an excellent system will be wrong sometimes, and it is necessary for students and teachers to be able to override any bad decisions; and secondly, a student's failure is hopefully not an inevitable event, and knowing about it is only useful when one knows what kinds of habits are leading to that negative outcome and what to do about them. Some machine learning algorithms are inherently more auditable

[15] https://files.semtech.athabascau.ca/public/TRs/TR-SemTech-11012011.pdf

[16] http://cohere.open.ac.uk

[17] http://oro.open.as.uk/25829/1/DeLiddo%2DLAK2011.pdf

[18] http://www.mindmeister.com/

[19] http://oro.open.ac.uk/31052/1/CCI%2DCSCWj%2D 2012preprint.pdf

[20] http://www.itap.purdue.edu/learning/tools/signals/

than others, which must inform decisions when designing such a predictive system. Even if a system is not capable of providing concrete suggestions for how students can modify their behavior, at least some feedback about why the system made a decision is enough to get students thinking about their own learning. Discussions between students and teachers about patterns in learning found by an artificial intelligence system would be pedagogically useful, regardless of whether humans agree with computers' assessments.

Course Signals is at present a somewhat blunt instrument, but does provide a valuable starting point for student introspection and dialogues with instructors about the learning process. Making predictions about student success is a great goal, but to provide these predictions directly to students, without more information about why predictions were made and how to avoid negative outcomes, could potentially cause problems. Some of the worst fears instructors have about learning analytics systems are that they are impersonal, that they will make mistakes and leave humans with no recourse, and that students will feel pressured to do things to "appease the system" rather than to improve their learning. In the case of Course Signals, some of these criticisms could perhaps be alleviated by giving these automated predictions to instructors, who can then decide for themselves if and when to intervene.

2 Applying Corpus Linguistics Techniques to Learning Analytics

2.1 What is Corpus Linguistics?

Corpus linguistics is a descriptive approach to linguistics which centers on observing and analyzing large, "naturally occurring" bodies of text. The motivation for this approach is that human language is extremely complex and generally does not strictly adhere to *a priori* handcrafted rules of grammar, discourse, etc. Rather than designing rules of language based on what linguists prescribe, it can be more informative to observe the way language behaves naturally in a collection of documents and use those observations to identify patterns, which will hopefully generalize to other documents not included in the study. Documenting and formalizing naturally-occurring patterns lies at the heart of all scientific endeavor: computational analysis of linguistic patterns marks the beginning of linguistics' jump from the humanities to "hard" science. Making linguistics computational makes it possible to automate tasks which previously could only be done by human beings, and to do so on a more massive scale than ever before.

A corpus (plural corpora) is a large collection of documents. How large the collection must be generally depends on the phenomenon under study, but, as with all scientific data collection, more data will lead to better results which will more aptly generalize to new situations. The notion of a "document" will also vary greatly by task. A document can be just a few words (for example, a Twitter update) or something larger (an essay or newspaper article). For speech applications, a document can even be audio data.

When selecting documents for a corpus, it is important to balance homogeneity with variety. For example, discoveries made when analyzing a corpus of political newspaper articles may or may not generalize to newspaper articles in other domains (sports, arts and leisure, etc.). If one means to restrict one's research to articles on politics, this is not a problem, but it would not be an optimal way of researching newspaper writing as a whole. Including articles on other topics may provide results that generalize more widely, but adding text from wildly different sources (Twitter, classical literature) may simply add noise to the data. The choice of documents in a corpus must be informed by the goals of the project, and generally should be fully representative of the kind of language under study. For example, a study on quality of student writing should include many A+ papers as well as many Fs, but would not need to include an equal representation of papers written by students from different parts of the United States or from different socioeconomic backgrounds.

2.2 What Can a Computer "Know" Using Corpus Linguistics?

Corpus linguistics techniques have been successfully applied to a wide variety of tasks. One example is part-of-speech tagging,[21] where a corpus that has been manually annotated with parts of speech

[21] http://www.aclweb.org/anthology-new/W/W96/W96-0213.pdf

is used to tune the parameters of an automated tagger such that it can accurately label text it has not seen before. A program which has been trained in this way can achieve extremely high degrees of accuracy when tagging unseen text, failing mostly on ambiguous words and words it did not encounter in training. Part-of-speech tagging helps not only to understand the syntactic structure of sentences, but also to make sensible inferences about meanings of words (e.g. "rock" in "sedimentary rock" vs. "rock the party") and sentences (which are built up compositionally from words according to their semantics and syntactic categories).

This sort of automated training of a program's parameters is called machine learning. Computer programs are, essentially, rules for a procedure for performing a task - here, humans concede that manually writing the rules for some tasks is just *too* difficult and time-consuming. Instead, programmers write rules for machines to "learn" their own rules of operation by receiving feedback from real-world data. Machine learning has become a vital tool for corpus linguistics, since even automatically compiled reports and statistics are too complex for humans to manually analyze in large corpora. Taking a machine learning approach, a computer program can find these patterns and modify its own behavior to make more accurate judgements.

While learning analytics often focuses on presentation of data in human-digestible format, corpus linguistics has trended towards creating systems which use data to automatically make decisions or assign labels. For example, a system which can label thousands of documents by topic will likely do so more efficiently than a human reader ever could, and can be used to quickly find documents of interest to researchers or clients. In corpus linguistics, these sorts of automated classifiers can be used to test linguistic hypotheses; in "industry" settings, they can be used to traverse huge data sets more efficiently that could be done manually (for example, checking for negative reviews of a product in twenty different languages).

Word-level tasks like part-of-speech tagging are important first steps in a variety of tasks which require robust "understanding" of document content and structure (for example, finding the antecedents of pronouns like "he" and "she"). However, document-level tasks are also possible. An example of a much more open-ended and difficult project is sentiment analysis (Pang, Lee and Vaithyanathan)[22] where a human-annotated corpus is used to train a program to differentiate between documents expressing generally positive and generally negative viewpoints (for example, movie reviews). A task such as this initially seems quite difficult for a computer, since determining the polarity of a review requires reading and understanding a document in its entirety, which computers are inherently incapable of doing. To make a task like this computational, the task must be converted into something computers *are* able to do, such as detecting the presence of words. Somewhat counterintuitively, Pang, Lee and Vaithyanathan achieved optimal accuracy in a sentiment analysis task by looking only at the *set* of words in a document, disregarding word frequency and context. This "bag of words" approach, when trained on enough data, can quickly and accurately gauge sentiment for large volumes of text (for example, all mentions of a movie or a political candidate on Facebook), giving a quick heuristic for general positivity/negativity towards a certain entity.

Computers are incapable of "knowing" anything, but they can infer quite a bit when given large amounts of data. As demonstrated in the Pang, Lee and Vaithyanathan paper, even quite shallow automated analysis of text (checking for presence of words, for example) outperforms classification rules handcrafted by humans by quite a bit, even when the humans use machine-generated statistics as an aid.[23] In contrast to trends in learning analytics, where distilled presentation of data is key, modern corpus linguistics generally recognizes that there are limits to what humans can do with metrics, and instead tries to use clever computational techniques to make inferences using real-world data.

A wide variety of machine learning algorithms can be applied to corpus linguistics: it is beyond the

[22] http://www.aclweb.org/anthology-new/W/W02/W02-1011.pdf

[23] Imagine having to produce, by hand, a categorization of all English words (approximately one million of them, not counting inflected forms like plural nouns, etc.) with an accurate *a priori* estimate of how likely they are to be used in a positive vs. a negative movie review. Even if one did take the time to do this, machine learning techniques would almost certainly achieve better accuracy.

scope of this paper to explain the inner workings and applications of such algorithms.[24] However, simpler and less powerful algorithms (e.g. naive Bayes classifiers) can often provide detailed statistics and decision trees explaining why they do what they do, which can be valuable in learning analytics for understanding why a student has been classified as "at-risk". More complex systems (e.g. neural networks) are capable of learning complicated, nonlinear relationships in data sets - an example might be socioeconomic background vs. GPA, which might have a very complex correlation - but are generally less auditable. For learning analytics applications, where auditability is likely a primary concern for reasons previously discussed, tradeoffs must be made.

Keeping machine learning in mind, let us return to the question of what computers can "know". An automated system can infer, to some degree, any judgement which humans can make in a relatively objective manner. Some possible example judgements are: classifying a word as a noun vs. a verb; distinguishing between different word senses ("neutron star" vs. "movie star"); labeling a movie review as positive or negative; judging a student paper to be well-written or poorly-written; classifying an argument as logically sound vs. flawed. If a human can build a corpus of documents with those judgements labeled, and those judgements objectively follow from information in the documents, then a computer can make an attempt at learning to make that classification.

Some judgements, however, are generally too complex for a machine to make without extra information about the text it is examining. Unlike humans, computers do not know anything about words they see apart from their representations in memory as bits, so making sound inferences about words without information about their semantic and syntactic properties is impossible. Providing the machine with informative "hints" to make learning possible is the goal of the annotation process.

2.3 The Annotation Process

Annotation is the manual addition of metadata to a corpus with the purpose of providing human insight into meaning. Making automated inferences about language is extremely difficult: the goal of annotation is to supply an automated system with extra information to make machine learning easier.

A real-world example of an annotation project is TimeML at Brandeis.[25] The goal of TimeML is to understand temporal relations in text, including ordering of events and links between them. To this end, human annotators label events with their semantic class, tense, aspect, etc. and temporal expressions with their category, duration, etc. A machine learning algorithm uses features extracted from this metadata to train itself and learn patterns. In the case of TimeML, a maximum entropy classifier, which uses linear regression to tune its own parameters, has been able to identify and make judgements about temporal relations with a high degree of accuracy. TimeML is a first step in automating understanding of the "flow" of time and events in narratives.

It is essential to have a clear goal in mind before starting an annotation task for many reasons - not the least of which is that human annotation, generally by temp workers specifically trained for the task, is time-consuming and expensive. Not only must the annotations be informative for a system learning the specific patterns of interest, but they must also be clear and objective enough to be applied consistently, so as not to add noise to the data and confuse the system. Human annotation is powerful because it allows a machine to get a taste of human intuition about language, but this must be done in such a way that an automated system can quantify and make sense of the results and use that information to make successful generalizations about data it has not seen before. Annotation is generally advisable for tasks (like document classification) in which handcrafted rules for solving the problem are too difficult to develop, and unannotated data lacks information a machine needs to make sense of the data (e.g. information about semantics, word classes, etc.)

Due to the tricky nature of problems that human annotation is designed to solve, the annotation process is often a cycle. First, an annotation schema and guidelines are devised and annotators attempt to apply them to texts, asking questions and clarifying rules. Generally, documents will be annotated

[24] Unlike learning analytics, the field of corpus linguistics has a comprehensive online repository of academic papers available for the technically inclined: http://www.aclweb.org/

[25] http://aclweb.org/anthology-new/P/P06/P06-1095.pdf

by more than one person in order to evaluate interannotator agreement, which is crucial if the phenomena under study are to be marked consistently enough for a machine to learn the patterns surrounding them. Once project leaders examine disagreements between annotators, they may make revisions to the annotation guidelines. Indeed, any problem difficult enough to require an annotation-based approach will be ambiguous to some degree, and some disagreements and confusion are to be expected.

Upon revising annotations and achieving sufficient interannotator agreement, an annotated corpus can be run through a machine learning algorithm. A large portion of the corpus will be used to train the system, while a smaller part will be set aside as unseen test data. If annotations are informative and consistent enough and training is successful, the system will achieve a high degree of accuracy on the unseen data. If not, revisions will need to be made, either to the annotation guidelines, or to the modeling of the problem itself.

In a learning analytics system, annotating all student data at a university by hand is obviously impossible, but there are many ways to address this issue. Corpus-based projects sometimes use "bootstrapping" tactics to use a relatively small amount of expertly-done human annotation to automatically annotate a larger corpus.[26] TimeML uses a grammar of handcrafted rules to preprocess data, tagging the "low-hanging fruit" beforehand and allowing annotators to move faster and focus on more difficult tasks. Obviously, a system which requires unseen documents to be annotated by hand to make judgements is pointless, since humans may as well make those judgements themselves - a system trained on annotated data is meant to "scale up" eventually and no longer rely on human annotations.

Of course, annotations of student work already exist in the form of graded assignments - not only in the grades themselves, but also in word-, sentence-, paragraph-, and document-level comments written by teachers or T.A.s. The difficulty lies in either getting this existing data in machine-readable format, or providing a convenient way to annotate assignments electronically. The previously discussed Cohere project is, in fact, a sort of annotation project in disguise. Cohere ingeniously induces students to annotate their *own* learning in ways that are fun and entertaining. One must only look to Facebook and Twitter to see how willing students are to provide (meta)data on themselves, as long as it is done transparently and helps them stay connected.

The "Point of Originality" project at Brandeis is one example of a task which might benefit from an annotation approach.[27] The original paper describes a way to use the WordNet[28] semantic hierarchy to estimate the degree to which a student has put an idea into his or her own words using a rule-based heuristic designed to agree with human intuition. It should be clear by now how someone with a corpus linguistics background might tackle this problem: by training a machine learning algorithm using a corpus of papers on which graders have made comments about originality in students' writing. Data generated about the corpus using WordNet would likely still be valuable in such an approach. Although this method would be more time consuming and expensive if annotated data is not already available, a system trained on such a corpus could possibly provide originality scores which were not dependent on specific query terms, and which might be more useful in cases where there was a close domain match between training and unseen documents - for example, an "originality" classifier specially trained on physics papers would have unique insight into what makes a physics paper original. A corpus-trained system might also be able to deal with technical terms like "heuristic" and "neutrino", which might not be covered by WordNet.

Corpus linguistics techniques could benefit learning analytics systems anywhere that large amounts of data are available. While human-annotated data is ideal, any sort of metadata (grades, demographic information, teacher comments, student evaluations) can be leveraged into annotations to help train a system. Corpora of student writing could gain insight into what makes student writing successful in terms of style, content, and rhetorical structure. Corpora of teacher evaluations could quickly compile lists of keywords for each teacher ("engaging", "boring",

[26] http://www-ssli.ee.washington.edu/people/bulyko/papers/TTS02.pdf

[27] http://bir.brandeis.edu/bitstream/handle/10192/23957/originality.pdf?sequence=1

[28] http://wordnet.princeton.edu/

"knowledgable"). A corpus of e-mails between faculty members would allow teachers to see and reflect on who is connected to whom and which academic relationships are most productive. Corpus-based analysis of university applications would allow objective analysis of what kinds of students are admitted and why, ensuring a fair admissions process. In the era of "big data", corpus-based analytics will be the only way to make sense of the world, and it behooves universities to lead the way.

3 Proposals for Practical Systems

3.1 Automated Flagging of Poor Student Writing

3.1.1 Background

For teachers in large "gateway" classes, grading student assignments can be time-consuming, and underperforming students may not get the feedback they need to modify their behavior until their grades have plummeted irreparably. A system which would automatically flag student writing which is likely to be poor shortly after it was uploaded to an LMS would *immediately* draw the attention of teachers and TAs to students who need extra help.

3.1.2 Proposal

We propose to use a collection of 100 student papers to train an automated system to distinguish between good and bad writing. Our approach is novel in that we propose to use teacher comments on graded papers as metadata to help the system learn what it means for writing to be good or bad. The system will be able to read in a paper it has never seen before and assign it a percent chance of being "bad", flagging the papers with the highest "badness" scores for human review. The goal is not to automate grading, but simply to alert teaching staff when a student may need timely intervention to avoid negative outcomes.

3.1.3 Summary

- flag poor writing automatically to enable quick intervention

- 100 student papers, 3-5 pages each

- machine-readable format, preferably plain text files

- grades, or binary classification of pass/fail - half the papers should be "good", half "bad"

- teacher/TA comments written legibly on printouts, or included digitally

- any available metadata on student authors (major, year) is a bonus

3.1.4 Costs

- one graduate student in computational linguistics working part time

- necessary software is open-source - no purchase necessary!

- if a large amount of teacher comments are available and are not available digitally, possibly a few hours data entry work by an undergraduate

- if an interface with an LMS is desired, someone familiar with the inner workings of the LMS could be brought on as a consultant

3.2 Analysis of Comments in Course Evaluations

3.2.1 Background

Most large universities likely have massive electronic archives of comments on course evaluations, but reading and making sense of all this text is infeasible without the aid of computers. Applying corpus linguistics techniques to this data would make it easy to administrators to traverse this data electronically and reveal trends and keywords associated with different courses and instructors.

3.2.2 Proposal

We propose to design a system for "exploring" the archives of course evaluations. The system will accept queries about courses and instructors and present output in easy-to-understand visual format. For example, a query about a specific course might provide a list of keywords which often appear in those course evaluations, which could be further broken down by instructor or by year. This is intended as an easy-to-use "dashboard" type system which could be used by administrators, faculty, or even students.

3.2.3 Summary

- easily, intuitively explore and analyze course evaluation comments

- digital text of course evaluations (as much as possible) complete with names of courses and instructors

- other data from the course evaluations is helpful - this would enable the system to output keywords from students who enjoyed the course vs. students who did not like it

- any other available metadata on the courses themselves (department, class size, average grade) is a bonus

3.2.4 Costs

- one computational linguistics student with experience in natural language processing

- one student familiar with graphical user interfaces to design the "dashboard"

3.3 Analysis of Edit History in Google Docs

3.3.1 Background

Google Documents already offers online collaborative editing of documents. This is an intuitive, convenient way to do group work remotely. Even if busy schedules do not allow collaborators to meet in person, they are able to keep up to date with others' edits and contribute when they have time. The Google Docs API could be used to create to generate metrics and visualizations for easy analysis of collaborations on group projects.

3.3.2 Proposal

We propose to design an online collaborative document editing system targeted towards group papers. This system would expand upon Google Docs in that it would present the development of a document as a series of "moves" (create, edit, delete) between collaborators. It would be possible to watch a playback of the evolution of a document, with detailed visualizations of which parts of the document were contributed by whom at what time. This is useful for keeping collaborators honest (in undergraduate settings, it is often the case that group work burdens are not shared equally), but is also invaluable for generating statistics which can show students how they work together: are they primarily contributors of new ideas? do they tweak the ideas of others? or do they just delete things they disagree with?

3.3.3 Summary

- visualize and analyze the life cycle of a document

- cloud-based collaborative editing

- builds on Google Docs framework

- intuitive, easy-to-navigate visualizations

3.3.4 Costs

- one or two students working part-time

- Google Docs API is free to use!

3.4 Mining of Staff E-Mails for "Connectivity" Analysis

3.4.1 Background

Analysis of connections between employees of an organization by looking at e-mails is a common first step in legal discovery. In fact, a corpus of e-mails from Enron is now popular among researchers in natural language processing, who have used it successfully as training data to analyze connectivity in social networks. Analysis of e-mails in university settings could provide an opportunity to objectively analyze which sorts of working relationships are productive, which ones are less so, and how "connected" community members are to each other.

3.4.2 Proposal

We propose to use faculty e-mails to generate visualizations of connectivity between staff. Many academic papers have recently become available on the topic of analysis of social networks: applying these same techniques to e-mails in academic systems could provide insights into who is most connected to the larger group, and who is more isolated. It may also be possible to do an analysis of the "character" of e-mails (e.g. offering help, rejecting a request). This could perhaps be done by manually annotating a small number of e-mails and then "bootstrapping" this data to do a rough annotation of the rest, or by clustering/topic modeling methods. Depending on the aims of the project, the data could be anonymized.

3.4.3 Summary

- analysis of connections between staff members
- need lots of e-mails (as many as possible), with a breakdown by department

3.4.4 Costs

- one part-time graduate student in computational linguistics
- one computer science student with experience in graphical user interfaces
- if desired, one part-time student to annotate e-mails by intent

3.5 Annotating Online Forums with Rhetorical Structure

3.5.1 Background

Online forums generally represent discourse in chronological order as a series of replies. However, as anyone who has used an online forum can attest, forum threads often get off topic, and not every contribution is equally valuable. The Cohere project (http://cohere.open.ac.uk/) seeks to disregard chronology and linear structure, representing online discourse instead as a graph of nodes, with each node a posting and the links between them "rhetorical moves" (e.g. disagree, provide evidence, introduce a new idea). A multi-dimensional web of ideas and concepts is evolved as students learn and communicate. This promising project is, at the moment, rather unique.

3.5.2 Proposal

We propose to develop a competitor for Cohere. There are huge advantages to evolving beyond the constraints of traditional online forums and representing learning as concept networks which students can annotate for themselves. A network-based representation more easily traversable and searchable for students reflecting on their own contributions and is more true to the non-linear nature of learning. It also has the fortunate side effect of producing data about cognition and rhetorical structure in learning. Our approach differs from that of Cohere in that we plan to use artificial intelligence to leverage this data into building a system which can *automatically* find rhetorical structure in documents and forums, and then construct concept networks which students can easily navigate, edit, and annotate with their own notes. Such a system would make it possible to make targeted searches of large numbers of unannotated documents based on rhetorical moves (for example, specific arguments which support global warming).

3.5.3 Summary

- non-linear representations of student discourse and learning
- rather than having a separate annotation phase, students annotate their own work - students must reflect on how their contributions fit into the larger discussion
- use this data to train an artificial intelligence system to annotate documents automatically and integrate them into concept networks
- university must take an active role in "marketing" the project to students and teachers - it only works if many students use it
- long-term, experimental project (5+ years)

3.5.4 Costs

- several experienced programmers specializing in cloud-based systems, web development, and graphical user interfaces
- at least one experienced programmer with a background in artificial intelligence/machine learning
- access to dedicated servers