Automatic Keyword Extraction for Learning Object Repositories

Kino High Coursey

University of North Texas and Daxtron Laboratories,Inc. kino@daxtron.com

Rada Mihalcea

University of North Texas**,** rada@cs.unt.edu

William E. Moen

University of North Texas, wemoen@unt.edu

The paper describes experiments in metadata generation for learning object repositories. Specifically, we present several methods for automatic keyword extraction and evaluate them on a collection of learning objects from an undergraduate history course. The results suggest that automatic keyword extraction is a viable solution for suggesting terms and phrases for metadata annotation.

Introduction

Learning object repositories are digital collections of educational materials (e.g., lectures, notes, presentations),which can be used to support learning. The main purpose of such repositories is to improve the sharing and reusability of the learning objects, which can be defined as “any digital resource that can be reused to support learning” (Wiley, 2000, p. 7). An important aspect of these repositories is the ease of accessibility to the constituent objects. To provide such access, learning objects in repositories typically have metadata records that provide information about salient features of the object and support discovery, selection, access and other functions. Metadata records generally result from manual creation processes. Such processes are both time consuming and expensive, and often represent an impediment against the scalability of the learning object repositories. For one overview of issues and opportunities for automated metadata generation, see (Greenberg, Spurgin & Crystal, 2005). In this paper, we explore automatic methods for metadata generation for learning objects. Specifically, we present several methods for automatic keyword extraction that can be used to assist or supplement manual assignment of subject terms, and evaluate them on a repository of history learning objects.

Our work is carried out within the broader framework of the Texas Course Redesign Learning Object Repository, a project in the Texas Center for Digital Knowledge (TxCDK), funded by the Texas Higher Education Coordinating Board (THECB). TxCDK has received funding from THECB to design and develop a learning object repository to store and make accessible the content from several first year undergraduate courses. In the first phase of research, we implemented a prototype repository, using content from a U.S. History I course. The hundreds of discrete learning objects that were derived from a single course presented challenges in metadata creation. In the second and current phase of the research, we are examining methods for machine-generated metadata that can assist the people who are otherwise manually creating metadata records associated with the learning objects.

The availability of automatically generated metadata in the form of keywords and associated conceptual labels may improve the discovery of potentially relevant learning objects by instructors in disciplines outside of the discipline for which the course was created. Automatically extracted keywords can be used as such to improve the search and discovery of learning objects, or can be used to assist or supplement the manual assignment of subject terms. Additionally, the keywords extracted from learning objects and the related concepts can be organized in ontological structures that can be used as an effective navigation tool through the repository. Ontology-like structures provide support for browsing through specification (e.g., suggest learning objects related to the more specific concepts of "Battle of the Atlantic" and "Liberation of Western Europe" in response to a query for "World War II") or generalization (e.g., suggest learning objects relevant to the more general concept "World Wars" or "Historical Events" in response to a search for "World War II").

We designed and implemented two techniques for automatic keyword extraction: (1) an unsupervised method relying on graph-based centrality algorithms, which we refer to as TextRank; and (2) a supervised method based on information drawn

from Wikipedia, which we refer to as Wikifier. Additionally, we implemented several hybrid methods that combine the two basic algorithms through reunion, intersection, and a method based on the longest common substring.

The paper is organized as follows: we first describe the collection of learning objects that we work with, and the manually generated annotations associated with this data set that are used as a gold standard to evaluate the performance of our methods. Next, we describe the TextRank and the Wikifier algorithms, followed by a description of the combination methods Union, Intersection, and Longest Common Substring (LCS). We then present the results of the evaluations on the learning objects data set, and conclude with a discussion of the results and directions for future work.

The Collection of Learning Objects

THECB funded the redesign of the U.S. history course as a part of a larger initiative, the Texas Course Redesign Project, which provides funding to redesign entry-level academic courses to improve student learning outcomes and lower costs using technology. The Center for Teaching, Learning, and Assessment at the University of North Texas carried out the redesign of the U.S. History I course. The course uses a variety of methods to present content including: HTML with embedded Flash objects, images, and video; MS Word documents; and PDF documents. Course content takes the form of lessons, self-assessments, interactive simulations, case studies, and other supporting materials.

The prototype learning object repository served as a proof-of-concept to demonstrate how a digital repository could be designed to store and provide user access to learning objects. In addition, the prototype work provided an opportunity to explore how an entire course could be decomposed into learning objects varying in granularity and complexity (e.g., compound objects comprising text, images, etc.). In addition to making the entire course available in the repository, we can also make component parts of the course available as discrete learning objects for reuse and repurposing.

For the prototype, we used the structure of the U.S. History I course to guide the decomposition of the content into four different levels of granularity:

- Units, comprising two or more lectures
- Lectures, comprising two or more topics
- Topics, in-context primitive learning objects
- Free-standing learning objects, which are primitive learning objects that may have value outside of the U.S. History course context.

From Federalism to Republicanism

In a brief period of time, the United States won its independence from Britain and created a new form of government, representing the culmination of decades of intellectual and political thought. Soon most Americans supported a republican-style government where the people were sovereign and they delegated power and authority to elected government leaders, although government leaders clashed as to the degree of direct control that the government should have over the people. In the midst of this argument a few citizen – for example, James Madison realized that the diversity of opinion over the government' structure was one of the strengths of the union and society taking shape and not a weakness.

From a Federalist paper about 1800. Federalists and Republicans are warring over the nation while Washington looks on from heaven.

At first there were no parties in the new nation and government leaders feared political parties, believing that they would tear the Republic apart. But factions formed around personalities, particularly those of Secretary of the Treasury Alexander Hamilton and Secretary of State Thomas Jefferson.

By 1793 Hamilton had built a network of supporters that emerged as a party called the Federalists. In reaction, Thomas Jefferson and James Madison built their own body of supporters, calling themselves the Republicans – eventually refined as the Democratic-Republicans. By 1796 the bi-partisanship in America's politics had emerged in full-force.

Another presidential election was set for 1800 and the contest between John Adams and Thomas Jefferson was heated as the differences that divided the Federalists and the Republicans became more pronounced. When the elector's votes were counted in February 1801, the Republicans came out on top, but by only a slim margin, 73 to 65.

Figure 1 An example of a learning object from the U.S. History I course

The entire U.S. History course content decomposed into nearly 300 separate learning objects, grouped into 81 topics. Figure 1 shows an example of a learning object from this course. Each of the learning objects was submitted into the digital repository together with a metadata record. One particularly challenging aspect of creating the records was assigning subject terms and creating summaries to represent what the learning object was about. To support the metadata creation, we developed an application containing a controlled list of subject terms that was linked to the metadata creation process for use

by the record creators. The subject terms in this list were derived from the text or were created by the domain experts to represent concepts in the text. One expert assigned metadata to each learning object. These manually assigned terms in the metadata records represent a good testbed for the evaluation of automatic techniques for metadata generation through keyword extraction. Appendix 1 shows the subject terms assigned by the domain experts for the learning object shown in Figure 1.

An important objective for the second phase of the project is to examine how automatic techniques can be used on the textual learning objects to assist metadata creation. The target of the automatic keyword extraction system is represented by the 81 topic-level learning objects, each containing relevant HTML, Flash, pictures and PDF text. All available text is associated with a topic-level learning object and provides the text processed by the keyword extraction methods. Associated with each topic-level learning object is an XML encoded Dublin Core (see http://dublincore.org/) metadata record that contains the subject terms manually assigned.

On average each topic contains 26404 words. The human-assigned subject terms in the records for each topic are used as the gold standard for the evaluation. There are 1178 subjects contained in the metadata records for the 81 topic-level learning objects and 708 (60%) of them are found in the text while 470 (40%) are not.

Related Work on Keyword Extraction

The state-of-the-art in keyword extraction is currently represented by supervised learning methods, where a system is trained to recognize keywords in a text, based on lexical and syntactic features. This approach was first suggested by Turney (1999), where parameterized heuristic rules are combined with a genetic algorithm into a system for keyphrase extraction (GenEx) that automatically identifies keywords in a document. A different learning algorithm was used by Frank (1999), where a Naive Bayes learning scheme is applied on the document collection, with improved results observed on the same data set as used by Turney (1999). Neither Turney nor Frank report on the recall of their systems, but only on precision: a 29.0% precision is achieved with GenEx for five keyphrases extracted per document, and 18.3% precision achieved with Kea (Frank 1999) for fifteen keyphrases per document.

In more recent work, Hulth (2003) applies a supervised learning system to keyword extraction from abstracts, using a combination of lexical and syntactic features, proved to improve significantly over previously published results.

TextRank

The first keyword extraction method that we use relies on an unsupervised graph-based ranking algorithm, which we refer to as TextRank (Mihalcea & Tarau, 2004).

Graph-based ranking algorithms

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of "voting" or "recommendation." When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

Formally, let G=(V,E) be a directed graph with the set of vertices V and set of edges E, where E is a subset of V x V. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it (predecessors), and let $Out(V_i)$ be the set of vertices that vertex V_i points to (successors). Using a ranking model inspired by PageRank (Brin & Page, 1998), the score of a vertex V_i is defined as follows:

$$S(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

where d is a damping factor usually set to 0.85.

Keyword extraction with graph algorithms

For the task of keyword extraction, the units to be ranked by the graph algorithm are sequences of one or more lexical units extracted from text, and these represent the vertices that are added to the graph.

Any relation that can be defined between two lexical units is a potentially useful connection (edge) that can be added between two such vertices. We are using a *co-occurrence* relation, controlled by the distance between word occurrences: two

vertices are connected if their corresponding lexical units co-occur within a window of maximum N words, where N can be set anywhere from 2 to 10 words.

The vertices added to the graph can be restricted with syntactic filters, which select only lexical units of a certain part of speech. One can, for instance, consider only nouns and verbs for addition to the graph, and consequently draw potential edges based only on relations that can be established between nouns and verbs. After experiments with various syntactic filters, including all open class words, nouns and verbs only, etc., the best results were obtained when using nouns and adjectives.

The TextRank keyword extraction algorithm is fully unsupervised, and proceeds as follows: first, the text is tokenized and annotated with part-of-speech tags – a preprocessing step required to enable the application of syntactic filters. To avoid excessive growth of the graph size by adding all possible combinations of sequences consisting of more than one lexical unit (ngrams), we consider only single words as candidates for addition to the graph, with multi-word keywords being eventually reconstructed in the post-processing phase.

Next, all lexical units that pass the syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of N words. After the graph is constructed (undirected unweighted graph), the score associated with each vertex is set to an initial value of 1, and the ranking algorithm described in the previous section is run on the graph for several iterations until it converges – usually for 20-30 iterations, at a threshold of 0.0001.

Once a final score is obtained for each vertex in the graph, vertices are sorted in reversed order of their score, and the top T vertices in the ranking are retained for post-processing. While T may be set to any fixed value, usually ranging from 5 to 20 keywords (e.g. Turney (1999) limits the number of keywords extracted with his GenEx system to five), we are using a more flexible approach, which decides the number of keywords based on the size of the text. For the data used in our experiments, T is set to a third of the number of vertices in the graph.

During post-processing, all lexical units selected as potential keywords by the TextRank algorithm are marked in the text, and sequences of adjacent keywords are collapsed into a multi-word keyword. For instance, in the text *Matlab code for plotting ambiguity functions*, if both *Matlab* and *code* are selected as potential keywords by TextRank, since they are adjacent, they are collapsed into one single keyword *Matlab code*.

Appendix 1 shows the keywords extracted by the TextRank method for the learning object shown in Figure 1.

Wikifier

The second keyword extraction method relies on Wikipedia – a free online encyclopedia, representing the outcome of a continuous collaborative effort of a large number of volunteer contributors. We use an unsupervised keyword extraction algorithm that works in two steps, namely: (1) candidate extraction, and (2) keyword ranking. This keyword extraction algorithm is part of the larger Wikify! system (Mihalcea & Csomai, 2007). Note that the reliability of the Wikipedia content is not relevant in our case, since we are only using the keyword annotations available in Wikipedia, and not the factual article content per se.

The *candidate extraction* step parses the input document and extracts all possible n-grams that are also present in the controlled vocabulary.

The *ranking* step assigns a numeric value to each candidate, reflecting the likelihood that a given candidate is a valuable keyword. While experiments were carried out with several ranking methods, the one that was found to work best is a measure based on Wikipedia "keyphraseness." This method exploits the vast information contained in the already annotated articles of Wikipedia. We estimate the probability of a term W to be selected as a keyword in a new document by counting the number of documents where the term was already selected as a keyword (count(D_{key})) divided by the total number of documents where the term appeared (count(D_W)). These counts are collected from all the Wikipedia articles.

$$P(keyword \mid W) \approx \frac{count(D_{key})}{count(D_W)}$$

This probability can be interpreted as "the more often a term was selected as a keyword among its total number of occurrences, the more likely it is that it will be selected again." Although this probability estimate could become unreliable for marginal cases where the counts are very low, as mentioned before, in our experiments we only consider the words that appeared at least five times in Wikipedia, which addresses this problem.

The keywords extracted by the Wikifier method for the learning object in Figure 1 are shown in Appendix 1.

Hybrid Methods

The two systems have different strengths that could be combined. TextRank provides an estimate of the attention a reader would give terms in any given text. Wikifier recognizes phrases that large numbers of people consistently use to point to Wikipedia articles. Wikifier uses the broad coverage of Wikipedia to recognize those entities that human annotators have felt were important to reference, while TextRank provides the ability to handle novel yet important topics for which no Wikipedia article may yet exist.

A problem does arise in using the output of both programs - the two extractors may generate different segmentations of the same element of text. TextRank utilizes extended noun phrase chunks identified by syntactic tagging, while Wikifier uses the anchor text that may be a common keyword or common object identifier. Thus TextRank will identify "Birth of Venus Sandros Botticelli" while Wikifier will identify "The Birth of Venus" and "Sandros Botticelli" as separate terms. To find commonality between the outputs of the two systems three methods were used: set union, set intersection union and longest common substrings (LCS).

Union

Union is simply the combination of words and phrases found in the output of either system. Each system's output is preprocessed into a set of phrases and component words. A phrase like "English Colonies" would result in "English", "Colonies" and "English Colonies" being added to the set. Thus if the manual annotation is "British Colonies", then at the word level "colonies" would match while at the phrase level it would not. A stoplist is used to prevent common closed class words from being added to the list.

Intersection

Intersection is simply the collection of words and phrases that are found in the output of both systems. A preprocess similar to that for union is performed before the set intersection is performed on both vocabularies.

Longest Common Substring

Finding the longest common subsequence (Rahman & Iliopoulos, 2006) is a common operation in genomics used to find the commonality between two DNA sequences and also to measure the syntactic similarity between terms used in different vocabularies or ontologies. Here we use LCS to extract the commonality between the vocabularies extracted by Wikifier and TextRank.

```
function LCSubstr(S[1..m], T[1..n])
L := array(0..m, 0..n)
z := 0  (length of longest match)
ret := {}  (set of longest matches)
for i := 1..m
  for j := 1..n
    if S[i] = T[j] then L[i,j] := L[i-1,j-1] + 1  (continue growing a matching chain)
    if L[i,j] > z  then z := L[i,j]   ret := {}       (new longest found)
    if L[i,j] = z  then ret := ret ∪ {S[i-z+1..i]} (an equal longest found)
return ret
```

Figure 2 Dynamic programming form of LCS Algorithm

The LCS problem is formulated as follows: given two strings S of length *n* and T of length *m*, find the longest string that is a substring of both S and T. A generalization of finding the common substring is finding the common subsequence, where a subsequence can 'skip' elements, but every element in the sequence occurs in order in both S and T. Both the common substring and subsequence can be found using dynamic programming, and the common substring case can be solved in O(*n***m*) using the algorithm in Figure 2 (Wagner & Fischer, 1974). In general n and m are less than 8 for the output of both algorithms. Applying LCS to the output of both algorithms allows coherent fragments of agreement to be found between both systems.

For each element in the Wikifier output, the LCS that can be found in all of the TextRank output for the same text is collected and added to list L1. Then for each element in the TextRank output the LCS is found in all the Wikifier output for the same text and added to list L2. The final output is the union of L1 and L2. This represents the longest fragments found in the vocabularies generated by both algorithms.

Experiments and Evaluations

Associated with each learning object is a set of manually assigned subject terms that describe what the learning object is about. This is taken as the gold standard defining an ideal set of keywords that should be applied to the object. However, one problem exists in that the terms in the subject definition may not exist in the text being analyzed or a synonym may be used. One example would be an annotator assigning "British Colonies" while the text only mentions "English Colonies." Another example might involve a number of component objects describing specific battles, but no string found in the available text mentions "Revolutionary War." Given that Wikifier and TextRank are both extractive, the performance is best measured relative to finding those subject terms that can be located in the text.

To evaluate the performance of the system at finding relevant keywords, the precision, recall and standard F-measure at both the word and phrase level are measured. Word level classification views both the subject terms and the text as bags-of-words. The goal is to be able to correctly classify a given word found in the text as either being a subject term or not. At the phrase level the number of correct and incorrect phrases generated relative to the number of human provided phrases that can be found in the text is measured.

$$precision = \frac{truePositive}{truePositve + falsePositive}$$

$$recall = \frac{truePositive}{truePositve + falseNegative}$$

$$fmeasure = \frac{2 * truePositive}{2 * truePositve + falsePositive + falseNegative}$$

Table 1: Phrase level classification

Method	**Precision**	**Recall**	**F-measure**
Wikifier	0.0323	0.7754	0.0620
TextRank	0.0357	0.2278	0.0618
Intersection	0.0746	0.2092	0.1100
Union	0.0209	**0.8389**	0.0408
LCS	**0.1275**	0.1867	**0.1515**

Table 2: Word level classification

Method	**Precision**	**Recall**	**F-measure**
Wikifier	0.0713	0.8586	0.1317
TextRank	0.1186	0.3956	0.1825
Intersection	0.2359	0.3005	**0.2643**
Union	0.0653	**0.8980**	0.1217
LCS	**0.2573**	0.2651	0.2611

Tables 1 and 2 show the results of the evaluation of the two individual methods (TextRank and Wikifier) and the three hybrid methods (union, intersection, and LCS). Of all five methods, LCS provides the best precision on both the word and phrase level, while as expected the union provides the highest recall. LCS provides the best overall F-measure at the phrase level, and is very close to the intersection method on the word level. However, for precision-based tasks LCS is preferred due to its relative performance at the phrase level. Overall, using LCS on the output of both TextRank and Wikifier provides a higher precision output than either alone.

It is also worth noting that the Wikifier alone provides a level of high-recall. This indicates that Wikipedia has been annotated in a way that is consistent with the intuition of the learning object annotators. To a large degree they both consider the same terms to be important and the Wikifier can identify these terms in text as relevant phrases.

Conclusions and Future Work

In this paper, we explored the use of automatic techniques for metadata generation for learning object repositories. Through experiments carried out on a collection of learning objects from an undergraduate U.S. History I course, we showed that a combination of keyword extraction techniques, bringing together graph-theoretical algorithms and methods based on knowledge derived from Wikipedia, can be successfully used to identify candidate keywords in learning objects. While the automatic methods are still error prone, their output can be useful to support and speed-up the task of the metadata creators For instance, the output of the Union method, which was shown to have high recall, can be used as an initial set of suggested keywords that can be further refined by the metadata creators. Alternatively, the more precise Intersection or LCS methods can be used to supplement existing sets of human-assigned subject terms.

One possible extension of the system is to use the information provided by the Wikifier to suggest keywords not in the text. Wikifier in addition to recognizing keywords used to reference articles provides the actual Wikipedia article titles. Given the set of relevant articles, the system can provide other terms used to refer to those articles, possibly addressing the synonym problem.

Another area of future research is to explore the effect of using the TextRank algorithm over the Wikipedia connectivity graph to find other articles that may be relevant to the topic. This should provide the ability to identify additional non-mentioned terms and possibly place the learning object in an overall context by finding the common articles related to the identified terms.

One issue in automatically evaluating the system is the fact that only a fraction of the annotator-selected subjects actually occurs in the texts. Additional evaluations will be needed to see how well the output of the system is relevant to user needs independent of the existing annotations.

ACKNOWLEDGMENTS

The Texas Center for Digital Knowledge and the project team gratefully acknowledge the funding from the Texas Higher Education Coordinating Board through the Texas Course Redesign Project, Grant #CR72105 - A Proof-of-Concept Repository for Learning Objects.

REFERENCES

Brin, S. & Page, L. (1998) *The anatomy of a large-scale hypertextual Web search engine*. Computer Networks and ISDN Systems, 30, 1-7.

Frank,E., Paynther, G.W. , Witten, I. H., Gutwin, C. & Nevil-Manning, C.G. (1999) Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*.

Greenberg, J., Spurgin, K. & Crystal, A. (2005). Final Report for the AMeGA (Automatic Metadata Generation Applications) Project. 2008. http://www.loc.gov/catdir/bibcontrol/lc_amega_final_report.pdf

Hauth, A. (2003) Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Japan, August.

Mihalcea,R. & Csomai, A. (2007) Linking Educational Materials to Encyclopedic Knowledge, In *Proceedings of the International Conference on Artificial Intelligence in Education* (AIED 2007)

Mihalcea, R. & Tarau, P. (2004) TextRank: Bringing Order into Texts, In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*

Turney, P. (1999) Learning to extract keyphrases from text. Technical report, National Research Council, Institute for Information Technology.

Rahman, M.S., & Iliopoulos,C.S. (2006) Algorithms for computing variants of the longest common subsequence problem. In Tetsuo Asano, editor, ISAAC, *volume 4288 of Lecture Notes in Computer Science*, 399–408. Springer

Wagner, R.A., & Fischer, M.J. (1974) The string-to-string correction problem, *Journal of the ACM*, 21, 168-173

Wiley, D.A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. 2008 http://www.reusability.org/read/chapters/wiley.doc.

Appendix

Gold standard of subject terms assigned and keywords extracted for the sample learning object from Figure 1.

Gold Standard Republicans, Democrat-Republicans, Political Parties, Thomas Jefferson, Jefferson Administration, Revolution of 1800, Judiciary Act of 1801, Strict Constructionism, Frederick JacksonTurner, Louisian Purchase, Britain, France, Impressment, Chesapeake, Embargo, War of 1812, James Madison, Madison Administration, War Hawks, Native Americans, American Indians, Tecumseh, Tenskwatawa, Pan-Indian Movement, William Henry Harrision, Tippecanoe, Non-Intercourse Act, Macon's Bill No.2, U.S.Navy, Canada, Andrew Jackson, Fort McHenry, Star Spangled Banner, Battle of New Orleans, Treaty of Ghent, Hartford Convention, James Monroe, Monroe Administration, Domestic Plolicy, Westward Expansion, Rush-Bago Agreement of 1871, Florida, General Andrew Jackson, Seminole, Panic of 1819, Slavery Debate, Missouri Compromise, Slavery, John Marshall, Dartmouth College v. Woodward, McCulloch v. Maryland, Gibbons v. Ogden, Early Nationhood, Era of Good Feeling
Wikifier United States, Federalism, Republicanism, independence, Britain, form of government, decades, republican, sovereign, James Madison, union, Federalist, 1800, Federalists, Republicans, Washington, heaven, the new nation, political parties, Republic, Secretary of the Treasury, Alexander Hamilton, Secretary of State, Thomas Jefferson, 1793, Hamilton, Jefferson, Madison, Democratic-Republicans, 1796, America, politics, full-force, presidential election, John Adams, elector, 1801
TextRank government leaders, Treasury Alexander Hamilton, Federalists, Secretary of State Thomas Jefferson, new form of government, Thomas Jefferson, United States History, new nation, few citizens, United States, James Madison, republican-style government, degree of direct control, brief period of time, own body of supporters, structure, votes, politics, for example, John Adams, political parties, Federalist paper, diversity of opinion, slim margin, presidential election, network of supporters
Intersection government, Federalist, States, politics, John Adams, presidential election, Federalists, James Madison, political parties, United, United States, Thomas Jefferson
Union 1801, republican-style government, supporters, Federalism, network, State, Britain, Treasury, United States History, Hamilton, brief period of time, government, direct, Federalist, Adams, elector, leaders, paper, election, diversity, Federalist paper, History, 1793, States, politics, Secretary of the Treasury, votes, John Adams, for example, parties, margin, period, political, new nation, presidential election, control, 1800, diversity of opinion, body, degree of direct control, opinion, republican-style, full-force, Democratic-Republicans, Secretary of State, few citizens, form, government leaders, Alexander Hamilton, Washington, Secretary, Republic, own body of supporters, Jefferson, citizens, the new nation, Federalists, slim, James, union, James Madison, slim margin, nation, degree, independence, John, political parties, decades, network of supporters, time, sovereign, Thomas, Republicanism, 1796, United, United States, form of government, heaven, Thomas Jefferson, Secretary of State Thomas Jefferson, Alexander, America, Madison, Treasury Alexander Hamilton, new form of government, republican, Republicans, presidential, structure
LCS political parties, new nation, Thomas Jefferson, John Adams, Secretary of State, Alexander Hamilton, James Madison, presidential election, United States, form of government