東京大学
情報理工学系研究科 創造情報学専攻
修士論文

差分対応付けによる多言語文書同期化支援システム

A System for Synchronizing Cross-language Documents with Differences Alignments

コ モンテン
Wenting Gu

指導教員　千葉 滋 教授

2013年1月

概要

本研究は複数人で多言語文書の同期化を効率的に実現することを目指し，異なる言語文書間の差分の対応付けにより，修正箇所を特定し，ユーザーに分かりやすく表示するシステム SynCLDoc を開発した．既存の多言語文書同期化支援ツールでは，オープンソースのソフトウェアのマニュアルのような各言語の版に独立した編集を加える多言語文書は，ある言語の版を修正したとき，他の言語の版の修正箇所の把握が困難であった。そこで，本研究は既存の文の対応付けアルゴリズムを利用し，多言語文書間の段落と文の対応関係を計算することで，異なる言語間の修正箇所を特定し，強調して表示するシステムを開発した．

段落と文の対応関係の正確さについて評価し，ソフトウェアのマニュアルのような多言語文書の段落の正確な対応率が 95% 以上であり，文の正確な対応率が 91% 以上であることを確認した。また，ユーザ実験を行い，初めに三つの言語がある多言語文書を同期化する場合，SynCLDoc を利用することで，必要な時間が従来のシステムを利用する場合より，約 1/3 削減することを確認した．また，一度同期化された多言語文書を修正し，再同期化する場合，SynCLDoc を利用することで，差分の特定時間を 80% 以上削減することを確認した．

Abstract

This paper describes a system named "SynCLDoc" which helps users to synchronize cross-lingual documents effectivity. SynCLDoc realized this function through locating corresponding areas of modifictions by calculating corresponding relationships between documents in different languages. SynCLDoc is developed for the reason that we found it is difficult for users to keep the contents of the cross-lingual documents (which can be added different contents to each language version) synchronized with existing tools for the reason that users cannot find corresponding areas of modifications easily. Therefore, we developed SynCLDoc, a system which can locate corresponding areas of modifications and show them to users through the relationships between paragraphs and sentences of documents in different languages calculated by existing sentence similarity algorithms.

We evaluated the accuracy of coresponding relationship linked by SynCLDoc. As a result, above 95% coresponding paragraphs are correct, and above 91% coresponding sentences are correct in cross-lingual documents such as software manual. Also, according to the usability testings, the time spent to synchronize documents which have three language versions by using SynCLDoc at the first time is 1/3 less than the time spent by using other existing tools. In addition, when some language versions of a synchronized cross-lingul documents are modified, SynClDoc reduces 80% of the time spent to locate corresponding areas of the modifications by using other existing tools.

目次

第 1 章

序論

本研究は，複数人での多言語文書の編集作業において，多言語文書の同期化を効率的に実現することを目的とし，多言語文書の同期化支援システム SynCLDoc （Synchronize Cross-Lingual Document）を開発した．本研究で論じている「多言語文書」は，複数の言語で記述され，各言語の版（例えば，日本語版，英語版，中国語版）の内容が同じであるべきであり，そして頻繁な変更が必要な文書と定義している．

SynCLDoc は既存の文の類似度アルゴリズムを利用し，多言語文書の各言語の版の段落と文の対応関係を計算することにより，異なる言語間の修正箇所を特定し，強調して表示することで，多言語文書の同期を支援するシステムである．ウェブアプリケーションとして実装しており，複数人が同時に利用可能である．本稿は， SynCLDoc の設計と実装について詳しく説明する．また，実装の評価結果及びユーザーによる実証実験の結果を踏まえ，システムの有用性を示す．

本章の節 1.1 では，多言語文書の同期化を支援するツールが必要の必要性及び既存の多言語文書同期化ツールを説明する．節 1.2 では，本研究の目的，研究手法及び意義を述べる．節 1.3 では，本稿の構成について述べる．

1.1 多言語文書の同期化を支援するツールの必要性

国際化が進んでいる現在では，情報共有のため，多くの文書が一つの言語での提供から複数の言語での提供になっている．例えば，ソフトウェアのマニュアルや Wikipedia [*1] のような百科事典ウェブサイトがあげられる．世界中の人々がインターネットを利用して，共同作業で文書を編集することによって，文書の多言語化の時間が短くなっている．また，文書を多言語化する過程も図 1.1（A） のような伝統的な作成と編集方法から図 1.1（B）のように変更する傾向がある．

伝統的な文書を多言語化する過程とは，ある言語でオリジナルな内容を書いて，（例えば，英語で書く），そして，それを他の言語に翻訳するものである．例えば，英語から日本語と中

[*1] http://www.wikipedia.org/

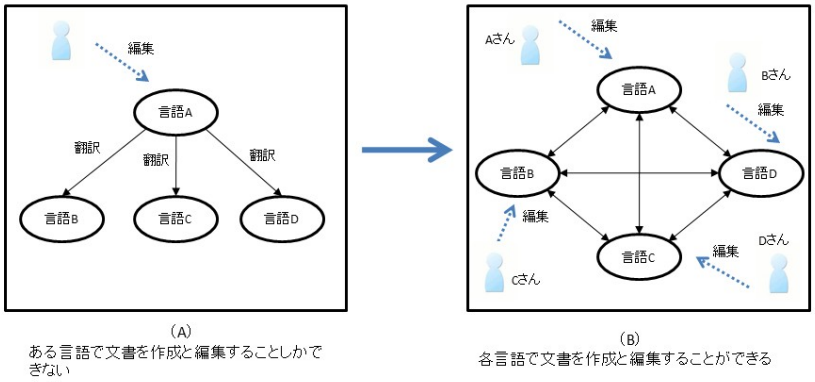

図 1.1. 伝統的な文書を多言語化する過程と多人数で協力的に文書を多言語化する過程

国語に訳す．また，文書を編集する時にも，オリジナルの言語（上記の例だと，英語）で書かれた内容しか編集が加えられず，他の言語（日本語と中国語）の内容はオリジナルの言語（英語）で書かれた内容の翻訳である．そのため，ユーザーは編集内容の対応箇所の探す手間がそれほどかからなく，翻訳支援ツールを利用したら，より効率的に多言語文書の内容の同一ができる．

しかしながら，現在は図 1.1（B）のような各言語の文書に対して独立した編集を行う事例が増加している．例えば， 2001 年の 1 月から 2012 年の 8 月までの約 10 年間に，Wikipedia に 285 種類の言語で 2,200 万の記事が作成され，それぞれの言語間で独立した編集が行われている．

このようにインターネットを利用し，協力的に文書の作成と編集する事例が増加している．特に，オープンソースのソフトウェアのマニュアルのような，頻繁な修正を与える多言語文書の各言語の版の作成と編集は複数人で独立して行われることが増えている．複数の人々が各言語の版に独立した編集内容を加えることで，多言語文書の各言語の版の内容が違うところが出てきる．例えば，先ほど述べた Wikipedia は記事[*2]の各言語ページの内容は大きな差分が存在する [1]．Wikipedia は，同じ記事の各言語ページは同じ内容を書く必要がないため，協力的な作業で文書の作成と編集は特に問題がない. それに対して，ソフトウェアのマニュアル，製品の操作マニュアルや小説などの最終的に各言語で書かれた内容を同期化する必要がある文章の同期を行うにあたっては，大きな問題点が存在する．つまり，本研究が論じる多言語文書の同期化が大きな問題点が存在する．

具体的には，同期を行うにあたって，ユーザーは多言語文書の各言語の版の修正内容及び，修正箇所が他の言語の版の対応箇所を分かりにくいため，多言語文書を同一することが困難で

[*2] ウィキペディアにおける「記事」とは、百科事典としての情報が記載されているページのことである。基本的には、標準名前空間にあるページが「ウィキペディアの記事」ということになり，百科事典で慣習的に使われている「項目」という言い方もされる。

ある（詳しくは章 2.1 参照）．言語の種類が多いほど，各言語版の内容や順番の差分が多いほど，同期化作業がより複雑になる．例えば，オープンソースプログラミング言語 Ruby [*3] は日本語と英語のマニュアルが独立に編集されていた．これらの二つの言語のマニュアルは相当差分があるため，手作業で同期化するのは困難であった．

この問題を解決するために，既にいくつかの多言語文書の同期化を支援するシステムが提案されている．例えば，Huberdeau らが提案した CLWE [2] というシステムは，多言語文書の同期化を支援する初めのシステムである．CLWE は多言語文書の各言語の編集履歴を記録し，ユーザーに各言語の文書の同期化の状態を示すことができるが，ある言語の版の内容を修正したら，この修正内容が他の言語の版のどこに対応しているかを編集者に指示しないため，ユーザーは修正の対応箇所を自分自身で探さなければならない． また，いくつかの新しい研究 [3, 4] でも多言語文書の同期化に着目したシステムは提案されているが，[3] は追加された内容だけの対応箇所を特定し，編集された内容や削除された内容の処理はしない問題や，[4] は Wiki ページの構造を依存し，Wiki ページの構造が存在しない文書に対応できないのは依然として問題となっている（章 2.2 参照）．

1.2 研究の目的と提案

本研究は，各言語の版の内容に独立した編集を頻繁に加える多言語文書を同期化するとき，編集内容の対応箇所の特定が難しい問題を解決し，より同期しやすい環境を提供することを目的とする．同一文書の異なる言語の版の差分の対応付けにより，修正内容と対応箇所を特定し，ユーザーに分かりやすく表示するシステムがあったら，複数人でより同期しやすい環境で多言語文書の同期化が実現できる．例えば，多言語文書の各言語の版の内容を段落と文単位で対応関係を取得することにより，ある言語の版の内容を修正する時，その修正内容に対応づけられた他の言語の版の正確な位置を探し，ユーザーにその修正内容及び対応する箇所を示す．

それを実現するため，本研究は，まず，既存の文の類似度計算アルゴリズムを利用し，段落と文の対応関係を計算する方法を提案する．段落の順番と文の順番なども含め，文の類似度計算結果により，多言語文書の異なる言語の版の段落及び文の対応関係を計算する．そして，言語 A,B に対する既に計算した対応関係 R(A, B) から，新たな対応関係を計算する方法を二つ提案する．一つ目は修正内容の対応箇所を古い対応関係から計算する．二つ目は，複数の言語の版 L1, L2, L3 について，すでに計算した対応関係 R(L1,L2)，R(L1,L3) から，新しい対応関係 R(L2,L3) を計算する．最後に，上記の方法で計算した対応関係を用いて，同期が必要な箇所をわかりやすく表示する手法を提案する．

本研究は，これらの提案を実際にシステムとして実装し，SynCLDoc と名付けた．SynCLDoc （図 1.2）は複数人が同時に利用可能になるため，ウェブアプリケーションとして提供しており，現在は英語，日本語，中国語の多言語文書の同期化を支援することが可能である．

SynCLDoc は各言語の版の内容に独立した編集を頻繁に加える多言語文書を対象として開

[*3] http://www.ruby-lang.org/ja/

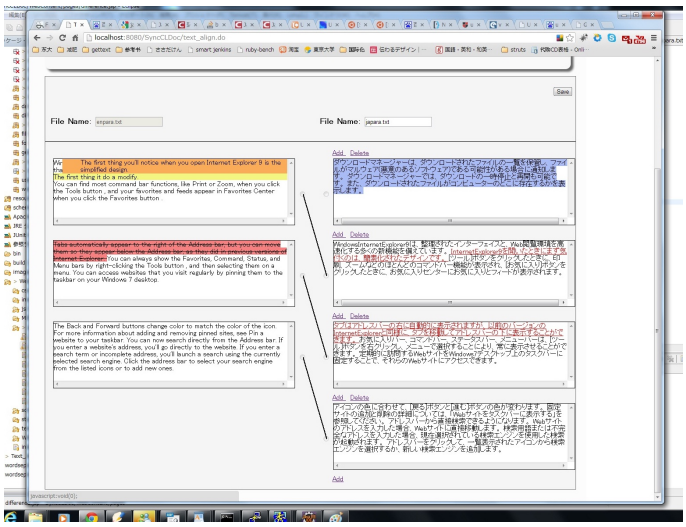

図 1.2. SynCLDoc システム多言語文書の同期化画面

発された．例えば，オープンソースのソフトウェアのマニュアル．Wikipedia のような各言語ページの内容は同じ内容を書く必要がない文章や，小説や記事などの修正頻度が少ない多言語文書は対象外である．

今後インターネットと国際化社会の発展によって，各言語の版の内容を頻繁に独立した編集を行う多言語文書がより多く存在し，これらの多言語文書の同期化作業のニーズも一層高まっていくと考えられる．従って，SynCLDoc の活躍が期待できる．また，本研究では [5] に提案された文の類似度計算アルゴリズムを利用して実装したが，今後文の類似度計算技術の発展することによって，SynCLDoc の正確さと効率も向上することが期待できる．

1.3　本稿の構成

本稿の構成は次の通りである．第 2 章では，多言語文書の同期化する現状にをまとめ，本研究の着目対象及び解決すべき問題について詳しく説明する．また，多言語文書の同期化ツールと翻訳支援ツールの相違点，既存の多言語文書の同期化する研究とその問題点について述べ，本研究がそれらの問題に対する提案と研究手法について詳しく述べる．第 3 章では，現状分析を踏まえ，本研究が提案するシステムの概要と実現手法について説明する．第 4 章では，開発した SynCLDoc の設計，利用手順とその実装について詳細に述べる．第 5 章では，多言語文書の各言語の版の対応関係の計算するアルゴリズム，他の言語ペアを活用した新たな対応関係の判定アルゴリズムを評価する．また，ユーザーによるシステム実験を行い，実験結果を分析した上で，システムの有用性にを議論する．第 6 章ではこれまでの考察，実験結果をまとまり，今後の課題についてを述べる．

第 2 章

多言語文書の現状と関連研究

本章では，多言語文書の編集やメンテナンスの現状を分析し，多言語文書の同期化の支援に関する既存研究を整理した上で，現在の多言語文書の同期化における問題点を提出し，多言語文書の同期化と従来の翻訳作業の相違点を明確にする．

2.1 多言語文書の編集における現状

国際化が進んでいる現在では，文書の多言語化の需要が日々高まっている．文書の多言語化を支援するために，多数の翻訳支援ツールが開発されている．これらの翻訳支援ツールは翻訳者がより高品質な翻訳を効率的に行い，文書の翻訳作業を支援している．翻訳支援ツールとして主なものは，翻訳ソフトと翻訳メモリツールである．

主な翻訳ソフトは機械翻訳技術を利用して実装されている．従って，翻訳ソフトの有効性は機械翻訳技術に大きな影響を受けている．翻訳ソフトの効率を向上するため，機械翻訳技術に関する研究が多数存在している [6, 7, 8, 9]．しかし，翻訳ソフトは用語辞書に強く依存し，用語辞書が存在しないと，翻訳の品質が低くなる．また，翻訳ソフトは，マニュアルなどの専門用語が多い文章では効果が高いが，小説のような文学性の強い文章や，ニュースなどの新語や固有名詞が多い文書は苦手である．

翻訳ソフトウェアの弱点を補強するため，翻訳メモリ技術が提出され，多くの翻訳メモリツールが開発されている [10, 11, 12, 13]．翻訳メモリの主な機能は翻訳者により書き起こされた翻訳を，その原文とともに，専用のデータベースに登録し，同じまたは類似の原文が出てきたときに自動的にデータベースから訳文を引用することである．従って，翻訳メモリツールは同じ文書を繰り返し翻訳する場合や類似した文章の翻訳における表現を統一する場合，翻訳者の時間と手間を減らし，文書全体としての翻訳品質の向上することができるものである．

現在の翻訳支援ツールは機械翻訳機能と翻訳メモリ機能が両方含まれていることは珍しくない．また，使いやすいユーザーインターフェイスを揃え，翻訳者により簡単に高品質な翻訳を

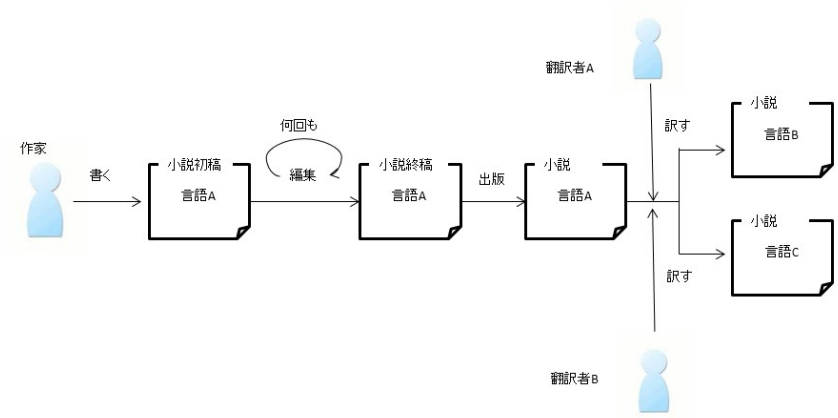

図 2.1. 小説の多言語化の流れ

行う環境を提供している [14, 15]．例えば，富士通の ATLAS [*1] や TRADOS [*2] などのソフトウェアである．

翻訳者が翻訳支援ツールを利用し，効率的に高品質な訳文を提供することができ，文書の多言語化作業は簡単になるため，従来の多言語文書の編集やメンテナンス作業に活躍している．従来の多言語文書の編集とメンテナンスは，ある言語の版の内容を編集し，他の言語の版の内容はこの言語の版の内容の翻訳である．いわゆる，ある言語で文書を書き，そして，この言語で書かれた文書しかオリジナルな修正ができない．

例えば，小説や一部のソフトウェアマニュアルなどの多言語化過程である．小説の多言語化は図 2.1 に示すように，まずはある言語で小説の原稿を書き，そして，原稿を何度も修正し，出版する．その次，翻訳者はこの小説を他の言語に翻訳する．一般的に言うと，出版された小説の編集が少ないため，一回だけの翻訳作業で終わる可能性が高い．また，他の言語の版は全部原文の翻訳なので，文書の節や段落や文の順番はほぼ同じである．そのため，原文に修正を与えても，翻訳者は少し時間がかかり，この修正が他の言語の版のどこに対応するか分かる．また，以前のソフトウェアマニュアルもある言語（主に英語）で書かれ，その他の言語に訳すことが多い．ソフトウェアマニュアルの多言語化過程と小説の多言語化過程が異なうところがある．それは，ソフトウェアマニュアルの編集は頻繁に行われる点である．オリジナルの言語で書かれた内容しか修正を与えられないソフトウェアマニュアルの同期化は小説の同期化より手間がかかるが，編集履歴で編集内容の確認ができ，各言語の版の内容と順番がほぼ同じのため，それほど困難でもない．翻訳者が翻訳支援ツールを利用し，翻訳作業を行うことで多言語化と同期化が実現できる．

しかし，節 1.1 で述べたように，インターネットの発展によって，現在多言語文書の編集作

[*1] 英日日英翻訳ソフトであり，Microsoft Word，Excel，PowerPoint，PDF や，メール，ウェブページなど多数のドキュメントの翻訳を支援するツールである．

[*2] 業界をリードする翻訳メモリソフトウェアである

業は複数人が協力的に行い，各言語の版に独立した編集を加える趨勢があるため，各言語の版の内容や順番も大きく違うこともある．特に，オープンソースのソフトウェアのマニュアルの作成と編集である．オープンソースのソフトウェアは世界各地の開発者が更新や改善ができるため，開発者は自分が慣れた言語の版のマニュアルしか編集しないことが多い．また，多くの人々が共同開発のため，オープンソースソフトウェアのマニュアルは頻繁な修正が行われる．その結果，オープンソースソフトウェアのマニュアルの各言語の版の内容は大きく違うことがある．節 1.1 に示した Ruby のマニュアルの例は一つの事例である．このような多言語文書の同期化作業を手作業で行うことが難しい．特に，言語の種類が多いほど，同期化作業がより複雑になる．

各言語の版の内容や順番も大きく違う多言語文書の同期化作業が困難である原因は下記の二つがある．

■原因 1　ユーザーは各言語の版の修正内容が差分かどうか分からないためである．

例えば，ユーザー A，ユーザー B，ユーザー C 三人がいており，多言語文書 D を編集する．

ユーザー A が多言語文書 D の言語 1 の版に修正 A を加えた．

ユーザー B が多言語文書 D の言語 2 の版に修正 B を加えた．

ユーザー C が多言語文書 D の言語 3 の版に修正 C を加えた．

多言語文書 D を同期化する時，ユーザーは修正 A，修正 B，修正 C の内容を取得し，この三つの修正内容は同じかどうかを確認しなければならない．修正 A，修正 B，修正 C が同じ修正の場合，同期化作業を行う必要がない．修正 A と修正 B が同じ修正で，修正 C が異なる修正の場合，言語 1 と言語 2 に修正 C を追加し，言語 3 に修正 A （または修正 B）を追加する必要がある．三つの修正がそれぞれと違う場合，各言語の版に他の二つの修正を追加する必要がある．

多言語文書の言語の種類が多いほど，各言語の版に修正の数が多いほど，各言語の版に与えられた修正が同じかどうかの判断が一層難しくなる．

■原因 2　ユーザーは各言語の版の修正内容が他の言語の版に対応する箇所が分からないためである．

図 2.2 に示すように，英語版と日本語版の段落の数が違い，各段落に差分がある．ユーザーはある言語の版の差分がある場合，他の言語の版の内容を一々比較しないと，この差分は他の言語の版にどこに対応するのか分からない．つまり，差分の対応する箇所の確定が多くの時間と手間が必要である．

これらの多言語文書の翻訳作業は翻訳支援ツールを利用し，効率的に行うことができるが，同期化作業を行う時，多言語文書の各言語の版の翻訳が必要な内容とその内容が他の言語の版のどこに対応するか特定できない．たとえ翻訳支援シールは文のアライメント機能が揃え，多言語文書の各言語の版の文の対応関係の取得ができる場合にも，多言語文書のある言語の版の内容を修正したら，ユーザーは時間と手間がかかり，その修正内容と他の言語の版の対応箇所を探さなければならない．その原因は，翻訳支援ツールの文のアライメント機能は，対応する原文と訳文を翻訳メモリデータベースに追加し，翻訳メモリ機能を作成し，再び原文に同じ文

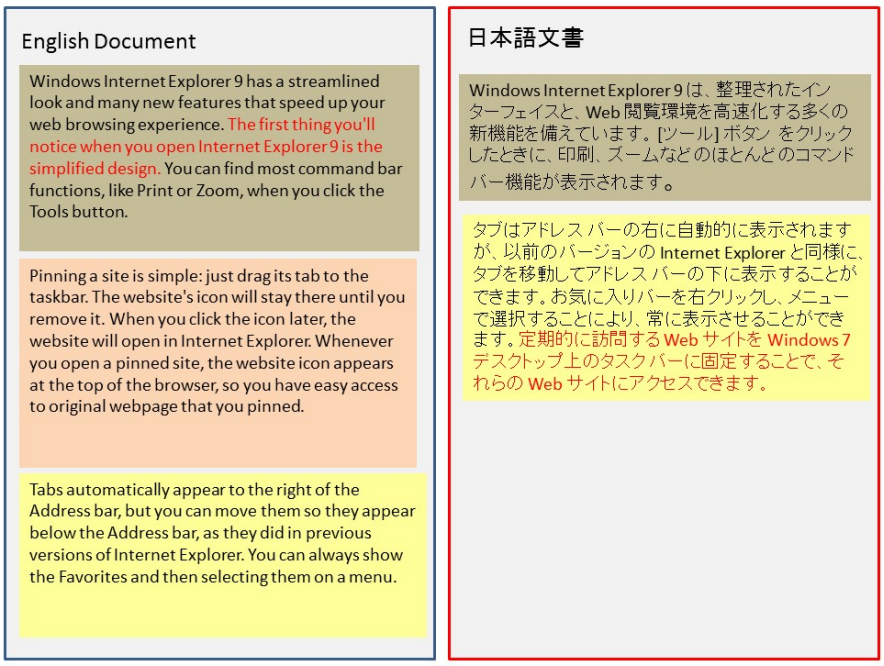

図 2.2. 修正内容の確認

を出てきた時に，翻訳メモリデータベースから訳文を取り出し，より速く，品質がいい翻訳を提供する機能である．

そこで，多言語文書の同期化を支援するツールが開発されている．多言語文書の同期化支援ツールは多言語文書の各言語の版の内容の差分を取り出し，また，ある言語の版の内容を修正すると，その修正内容の箇所を取り出し，そして他の言語の版のどこに対応するかを特定し，ユーザーに分かりやすく表示するツールである．多言語文書の同期化を支援するツールはユーザーが同期化作業における修正の対応箇所を特定する手間を減らすことで，より同期しやすい同期化環境を提供するものである．

2.2　関連研究

多言語文書の同期化を支援する関連研究がいくつか存在する．

Huberdeau らが提案した CLWE [2] は多言語文書の同期化を支援する初めのシステムである．CLWE は多言語文書の各言語の編集履歴を記録し，ユーザーに各言語の版の状態を示す．図 2.3 は CLWE の画面である．図 2.3 に示すように，CLWE はある言語の版の内容を修正すると，他の言語の版の状態は最新ではなく，更新が必要になるメッセージが出る．また，編集画面の上部に修正内容を表示する．これにより，ユーザーは多言語文書の各言語の版の状態がすぐに分かることができる．また，編集画面に修正内容を直接確認でき，対応しやすくなる．

ユーザーは CLWE を利用することで，多言語文書の同期化状態の確認と修正内容を取り出す手間を減らすことができるが，その修正内容は他の言語のどこに対応しているかが特定でき

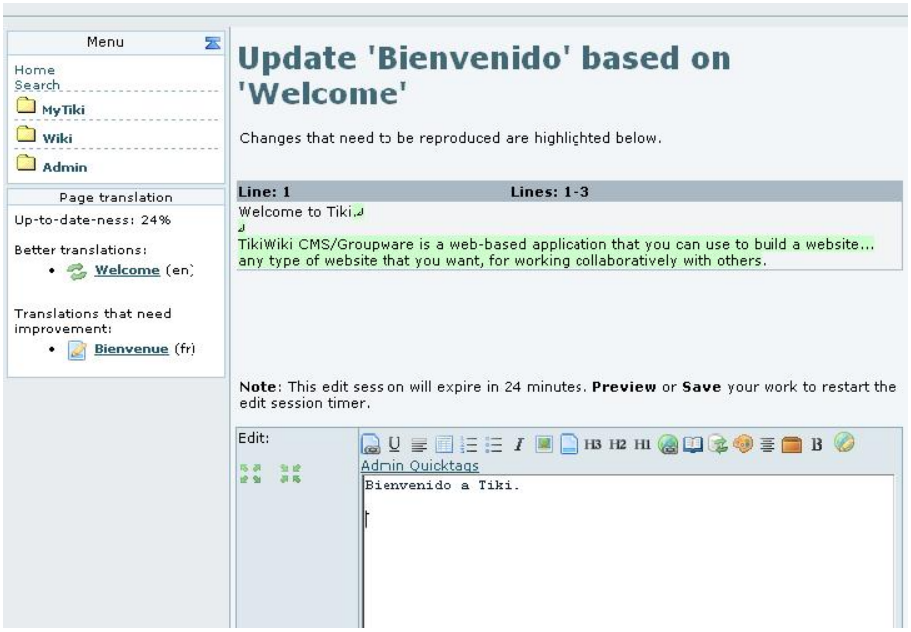

図 2.3. CLWE システム画面

ないため，対応箇所を探さなければならない．また，CLWE は各言語の版の修正が同じかどうか判断できず，ユーザーに間違って更新知らせが出る場合もある．例えば，ユーザー A は多言語文書の英語版の一段目の初めに「This is a test.」という文を追加する．ユーザー B は日本語版の一段落目の初めに「これはテストです.」という文も追加する．その場合，英語版と日本語版に同じ修正を与えれたため，多言語文書の同期化を行う必要はないが，CLWE はこの判断を行うことができず，ユーザーに多言語文書の英語版と日本語版が更新が必要というメッセジーを表示してしまう．

本研究は，多言語文書の各言語の版に与えた修正が同じものかどうかを判定し，多言語文書の同期化状態をユーザーに示すほか，各言語の版の修正内容と他の言語の版の対応箇所を特定し，表示するシステムを提案している．

[3] は Wikipedia の記事の各言語ページの内容を同期化するフレームワークを提案している． [3] は文の類似度と前後関係によって，ある記事に対する各言語の版の Wikipedia ページ内の新しい内容を取得し，文間の距離から追加された文の対応箇所を特定し，この新しい内容の他の言語ページの対応箇所を計算することを提案し，実装した．確かに，著者たちは文の類似度によって文の対応関係をつけ，追加された情報の上下の文の関係で他の言語内の対応箇所を特定する提案もしたが，対応箇所の特定の正確率はさらに低くため，フレームワークに利用しなかった．

[3] の評価によると，ある言語の版のページに文を追加すると，追加された情報が他の言語の Wikipedia の対応箇所の特定の正確率は以下である．彼らは一つの Wikipedia の記事に対して，ある言語のページに文を追加すると，他の言語のページにこの追加された文の対応箇所を特定する．特定された文の対応箇所は正しい節にある結果は約 83% であり，特定された文

の対応箇所は正しい段落にある結果は約 75% である．

しかし，著者たちは特定された文の対応箇所と原文の追加された箇所は一致するのか評価しておらず，文単位で対応箇所の正確率は未知である．また，[3] は追加された情報だけの対応箇所を特定し，修正された内容や削除された内容などの処理は明記しなかった．我々は，追加された情報だけではなく，修正された内容や削除された内容の対応箇所を文単位で特定するシステムを提案し，実装を行った．

CoSyne [4] は Wikis の同期化をするフレームワークとして提案された．CoSyne は Wiki ページを構造解析することで差分をとり，機械翻訳による訳文を Wiki ページに追加する．しかし，構造解析による差分をとるのは Wiki のページ構造に依存があり，マニュアルや小説や記事など Wiki ページではない文書に対応できない．また，CoSyne はまだ開発中で，著者たちによる評価はないため，フレームワークの有用性は判断できなかった．本研究は，同期化に対する要求が強くない Wiki ページを対象とはしておらず，分かりやすく書かれたマニュアルのような多言語文書を対象として，同期化を支援するシステムを提案し，実装まで実現した．

第 3 章

提案

複数人で協力して多言語文書の各言語の版を独立に修正すると，多言語文書の各言語の版の内容の差が生じる．言語が異なるため，ユーザーは簡単に多言語文書の各言語の版の差分とその差分の他の言語の版の対応箇所をすぐに見つけることはできないため，多言語文書の同期化が困難である．特に，オープンソースのソフトウェアのマニュアルのような頻繁に修正を与えられる多言語文書は，各言語の版の内容が大きく違い，より同期しにくくなる．

そこで，本研究はソフトウェアマニュアルのような頻繁に修正される多言語文書を対象にし，複数人での上記のようの多言語文書の同期化作業を支援し，より同期しやすい環境を提供するシステムを開発することを目的とする．本研究は異なる言語文書間の段落と文の対応付けにより，修正内容と対応箇所を特定し，ユーザーに分かりやすく表示する手法を提案する．本研究の対象はソフトウェアのマニュアルのような頻繁に修正され，各言語の版の内容を同期する必要がある文書である．小説のようなオリジナルの言語の版しか修正を与えなく，修正頻度も少ない多言語文書や Wikipedia のような各言語ページの内容は同じではなくても構わない文書は本研究が提案したシステムの対象外である．

本研究は文の類似度計算アルゴリズムを基づいて，多言語文書の各言語ペア間の段落の対応関係と段落毎の文の対応関係を取得し，計算された対応関係から各言語の版の内容の差分を判断する．ある言語の文書を修正するとき，修正された文（追加，変更，削除）の対応関係から多言語文書の各言語の版の状態を判断し，修正内容と対応箇所の特定を行う．また，類似度計算アルゴリズムによる対応関係の取得の誤差を減らすため，既存の言語ペア間の対応関係から新しい言語ペア間の対応関係を計算する．もちろん，既存の類似度計算アルゴリズムによる取得された段落と文の対応関係が完全に正確ではないため，ユーザーは手作業で対応関係の修正ができるようにする．

本章では，まず，類似度計算アルゴリズムから多言語文書の二言語間対応関係を計算する方法を章 3.1 で述べる．また，他の言語ペアを活用した新たな対応関係の判定方法を章 3.2 で述べる．そして，一度対応付けた言語の版の内容を修正したら，計算された対応関係から多言語文書の各言語の版の状態と対応箇所を特定する方法については章 3.3 述べる．

3.1　類似度アルゴリズムを利用した二つの言語の版の対応関係の計算

多言語文書の各言語の版の差分とその差分が他の言語の版のどこに対応するかという箇所をユーザーに提供できると，より単に多言語文書が同期化できると考えられる．本研究は任意の文の類似度計算アルゴリズムを応用することで，多言語文書の各言語の版の内容の相違点を判定し，各言語の版の対応関係が取得できると考えた．また，文書では，似ている段落や文が重複存在する場合もあるので，多言語文書の文と文の類似度による対応関係の計算のみでは不十分である．段落間の類似度，段落と段落の順番及び文と文の順番なども考えなければならない．そこで，本研究は段落の対応関係を計算する方法（節 3.1.1）と対応付けた段落間の文の対応関係を計算する方法（節 3.1.2）を提案する．

3.1.1　段落の対応関係の計算

多言語文書の二つの言語の版の対応関係を計算するため，この二つの言語の版の段落間の対応関係を取得する必要がある．

多言語文書 D の言語 A の版と言語 B の版の対応関係を計算する場合，$Pa = pa_1, pa_2, ..., pa_m$ は多言語文書の言語 A の版の段落の集合であり，$|Pa|$ は言語 A の版の段落の数であり，段落 pa_i は言語 A の版の i 番目の段落である．つまり $pa_i \in Pa$ である．同じように，$Pb = pb_1, pb_2, ..., pb_n$ は多言語文書の言語 B の版の段落の集合であり，$|Pb|$ は言語 B の版の段落の数であり，段落 pb_j は言語 B の版の j 番目の段落である．つまり $pb_j \in PB$ である．$LPa_i = l(pa_i, pb_1), l(pa_i, pb_2), ..., l(pa_i, pb_n)$ は言語 A の版の i 番目の段落と言語 B の版の段落の類似度関係の集合である．類似度関係 $l(pa_i, pb_j)$ は段落 pa_i，段落 pb_j，と類似度 $sim(pa_i, pb_j)$ がある． つまり，$l(pa_i, pb_j) = pa_i, pb_j, sim(pa_i, pb_j)$ である．

本研究は言語 A の版と言語 B の版の段落の対応関係の集合 Rab を計算するため，任意の文の類似度計算アルゴリズム $SIM(S1, S2)$ を利用して，段落と段落の類似度 $sim(S1, S2)$ を計算し，対応関係を付けるアルゴリズム $ParaRelation(Pa, Pb)$（Algorithm1）を提案する．まずは，言語 A の段落の順番で，段落 pa_i を取り出し，言語 B の段落 $pb_j (0 < j < n)$ との類似度 $sim(pa_i, pb_j)$ を計算する．もし，$sim(pa_i, pb_j)$ が σ （0 ～ 1 の値）より大きい場合，段落 pa_i と段落 pb_j を対応付け，段落の対応関係 Rab にこの段落の対応関係 $r = [pa_i, pb_j]$ を追加する．そうではない場合，$l(pa_i, pb_j) = pa_i, pb_j, sim(pa_i, pb_j$ を集合 LPa_i に追加する．最後に，段落の類似度関係 LPa_i 集合から最も適切な対応関係を取り出す $Best(LPa_i)$ アルゴリズムを利用する．$Best(LPa_i)$ アルゴリズムは LPa_i の中に存在する段落 Pa_i と言語 B の版の段落の類似度の中，最大の類似度が Γ （0 ～ 1 の値）よ小さい場合，段落 pa_i に対応する段落は存在しないとする．そうではない場合，段落 Pa_i と段落 Pa_{j} の類似度と二つの段落の位置を考えて，対応関係を付ける．

しかし，システムによる自動的に段落の対応関係の計算は一対一の段落の対応関係しか計算

Algorithm 1 二つの言語の版の段落間の対応関係を計算するアルゴリズム

アルゴリズム $ParaRelation(Pa, Pb)$:
begin:
for each $pa_i \in Pa$ **do**
  $LPa_i = \emptyset$;
  for each $pb_j \in Pb$ **do**
    if pb_j は対応付けていない **then**
      if $SIM(pa_i, pb_j) > \sigma$ **then**
        $R \leftarrow [pa_i, pb_j]$;
      else
        $LPa_i \leftarrow [pa_i, pb_j, sim(pa_i, pb_i)]$;
      end if
    end if
  end for
  $R \leftarrow Best(LPa_i)$;
end for
end

できない．これは，いくつかの既存の多言語文書の内容を参照し，段落の対応関係は 1 対 N の状況が多くないこと, かつ計算量を減らすために決めた．もちろん, 多言語文書の各言語の版の段落の対応関係は一対一ではない場合も存在するため，アルゴリズム $ParaRelation(Pa, Pb)$ で段落の対応関係を指定することが不十分である．本研究では，多言語文書の二つの言語の版に一対多の段落の対応関係が存在したら，ユーザーは手作業で段落を分けて，段落の対応関係を一対一にする．

3.1.2 文の対応関係の計算

本研究は，任意の文の類似度計算アルゴリズム $SIM(S1, S2)$ を利用して，二つの言語の版の文の対応関係を計算する．既存の文の類似度計算アルゴリズムは多数があり，その中の効率が良いものをシステムに投入する．また，二つの言語の版の全部の文の類似度から対応関係を付けるではなく，対応付けた段落の文の対応関係のみをつける．節 3.1.1 の段落の対応関係を計算するアルゴリズムによる計算された段落の対応関係を利用し，文の対応関係を計算する．段落の対応関係は一対一しか計算できないが，文の対応は 1 対 1，1 対 2，1 対 3，2 対 1，3 対 1 が許す．

対応付けた段落の文の対応関係を計算するアルゴリズムは Algorithm2 に示すように，R は計算された段落の対応関係 r の集合であり，$|R|$ は段落の対応関係の数である．px_i は言語 x の i 番目の段落であり，$|px_i|$ は言語 x の i 番目の段落の文の数である．Sx_{ij} は言語 x の i 番目の段落の j 番目の文である．Rs は段落の文の対応関係の集合である．$SIM'(A_i, B_j)$ は

Algorithm 2 対応付けた段落の文の対応関係を計算するアルゴリズム

```
SenRelation(R) :
begin:
for each r_i ∈ R do
  pa_i, pb_i を r_i から取り出す;
  for each Sa_ix ∈ pa_i do
    for each Sb_iy ∈ pb_i  do
      if Sb_iy は対応付けていない then
        if  SIM'(Sa_ix, Sb_iy, Rs) then
          break;
        end if
      end if
    end for
  end for
end for
end
```

文 A_i と文 B_j の文とその前後の 2 文含めて対応関係を計算するアルゴリズム（Algorithm3）である．sim_n は類似度を計算する二つの文の情報と計算された類似度を持つ変数である．$MAX(sim_1, sim_2, sim_3, sim_4, sim_5)$ は sim_1 から sim_5 の中に，最も大きい類似度が存在する $sim_i(0 < i < 6)$ を返すアルゴリズムである．

3.1.3　差分の判断

多言語文書の二つの言語の版の対応関係を付けたら，この二つの言語の版の差分が簡単に分かる．段落や文が対応関係に存在しない場合，この段落や文は差分である．類似度アルゴリズムによる計算された対応関係が完全に正確ではないため，ユーザーはシステムによる計算された対応関係の確認と修正が必要である．

当然，文化的な理由により，多言語文書のある言語の版だけに存在する段落や文などがある可能性がある．或いは，いくつかの言語の版に存在するが，他の言語の版には存在しない内容なども考えられる．本稿では，これらの内容を「特殊段落」と「特殊文」と呼ぶ．システムによる自動的に特殊段落と特殊文の認識は行わず，ユーザーが指定する必要がある．ユーザーが特殊段落と特殊文を指定したら，システムはこの情報を対応関係に記録し，特殊段落と特殊文を差分ではないと扱う．

Algorithm 3 一対多（最大一対三）の文の対応関係を計算するアルゴリズム

アルゴリズム $SIM'(A_i, B_j, Rs)$:
begin:
$sim_1 < -[A_i, B_j, SIM(A_i, B_j)]$;
$sim_2 < -[A_i, B_j B_{j+1}, SIM(A_i, B_j B_{j+1)}]$;
$sim_3 < -[A_i, B_j B_{j+1} B_{j+2}, SIM(A_i, B_j B_{j+1} B_{j+2})]$;
$sim_4 < -[A_i A_{i+1}, B_j, SIM(A_i A_{i+1}, B_j)]$;
$sim_5 < -[A_i A_{i+1} A_{i+2}, B_j, SIM(A_i A_{i+1} A_{i+2}, B_j)]$;
if $MAX(sim_1, sim_2, sim_3, sim_4, sim_5) > \xi$ **then**
  $Rs \leftarrow MAX(sim_1, sim_2, sim_3, sim_4, sim_5)$;
  return true;
else
  return false;
end if
end

3.2 他の言語ペアを活用した新たな対応関係の判定

類似度アルゴリズムによる計算された対応関係は誤差があるため，ユーザーの確認作業が必要である．従って，言語の種類が多いほど，ユーザーの確認作業は多くなる．例えば，日本語，英語と中国語三つの言語の版がある場合，ユーザーは日本語版と英語版の対応関係，日本語版と中国語版の対応関係，中国語版と英語版の対応関係を確認しなければならない．

そこで，本研究は，ユーザーの確認作業を減らすために，他の言語ペアを活用した新たな対応関係の判定方法を提案する．先の例を見ると，ユーザーは英語版と日本語版の対応関係，英語版と中国語版の対応関係を確認したら，この二つの対応関係は正しいと認める．そして，英語版を仲介として，日本語版と中国語版の対応関係の取得することができる．

下記の三つのステップで，他の言語ペア（英語と日本語，英語と中国語）を活用した新たな対応関係（日本語と中国語）を判定できる．まずは，段落の対応関係を判定する．

1. 日本語段落 Pj が英日対応関係 Rej に存在する場合
   Rej から，Pj が対応する英語段落 Pe を取得し，Pe は英中対応関係 Rec に存在するかどうかを判断する．存在する場合，Rec から，Pe が対応する中国語段落 Pc を取得し，Pj と Pc を対応付ける．存在しない場合，Pj は差分である．
2. 中国語段落 Pc が英中対応関係 Rec に存在する場合
   同じように，Rec から，Pc が対応する英語段落 Pe を取得し，Pe は英日対応関係 Rej に存在するかどうかを判断する．存在する場合，Rej から，Pe が対応する日本語段落 Pj を取得し，Pc と Pj を対応付ける．存在しない場合，Pc は差分である．

3. 日本語段落 Pj が英日対応関係 Rej に存在しない場合と中国語段落 Pc は英中対応関係 Rec に存在しない場合
   節 3.1.1 のアルゴリズム $ParaRelation(Pa, Pb)$ で対応関係を計算する．

また，各段落の文の対応関係も上記の三つのステップと同じように取得できる．

3.3　多言語文書の各言語の版の状態と修正箇所を特定する方法

本研究は，多言語文書の各言語の版の段落と文の対応関係を付けたら，多言語文書を修正すると，対応関係から各言語の版の状態，修正内容，及び修正内容は他の言語の版のどこに対応するかを特定する方法を提案する．

修正内容が追加，変更と削除の三種類を定義した．初めに二つの言語の版を対応関係を付けるとき，差分と判定された段落や文は全部追加されたものとして扱う．対応関係が存在する多言語文書のある言語の版を修正したら，システムは既存の同じ言語の差分を取る技術を利用し [16, 17, 18]，修正内容を取り出し，修正種類を判断する．そして，下記のステップで修正内容の対応する箇所を特定する．

1. 修正種類が追加の場合
   追加された段落や文の追加箇所が対応関係に存在しない場合，追加された文の追加箇所は元々差分なので，他の言語の版の状態の変化がなく，再び差分の対応箇所を特定する必要もない．追加された段落や文の追加箇所が対応関係に存在する場合，他の言語の版に，同じ追加箇所にこの修正があるかどうかを判断する．同じ修正がある言語の版は，状態が最新であるが，同じ修正がない言語の版は，状態は古くなる．他の言語の版の対応する追加箇所に，追加が必要というメッセージをユーザーに表示する．
2. 修正種類が変更の場合
   変更された段落や文は対応関係に存在しない場合，この変更の原文は差分なので，他の言語の版の状態の変化がなく，再び差分の対応箇所を特定する必要もない．変更された段落や文は対応関係に存在する場合，他の言語の版に，同じ修正があるかどうかを判断する．同じ修正がある言語の版は，状態が最新であるが，同じ修正がない言語の版は，状態は古くなる．そして，対応関係から対応する段落や文の箇所を取り出し，ユーザーに表示する．
3. 修正種類が削除の場合
   削除された段落や文は対応関係に存在しない場合，この修正は新しい差分が生じないので，他の言語の版の修正が必要ではなく，多言語文書の各言語の版の状態が最新である．削除された段落や文は対応関係に存在した場合，他の言語の版に，同じ修正があるかどうかを判断する．同じ修正がある言語の版は，状態が最新であるが，同じ修正がない言語の版は，状態は古くなる．そして，対応関係から対応する段落や文の箇所を取り出し，ユーザーに表示する．

もちろん，特殊段落と特殊文（節 3.1.3 参照）を修正を与える時，上記の判断方法とは少し違う．ユーザーはある段落や文を特殊段落や特殊文と指定したら，この特殊段落や特殊文を修正しても，削除しても，この特殊段落と特殊文が存在しない言語の版に対して，影響を与えない．つまり，新しい差分が生じないことである．一方，この特殊段落と特殊文が存在する言語の版に対して，上記の判断方法で文書の状態と対応箇所の特定を行う．

第 4 章

SynCLDoc の設計と実装

本章では第 3 章で提案した手法をウェブアプリケーションとして提供するように実装した SynCLDoc の設計と各構成部分の実装について述べる． SynCLDoc は Java でウェブアプリケーションとして実装され（ソースコードは約 10,650 行がある），複数人で協力的に多言語文書の同期しやすい環境を提供し，多言語文書の同期化を支援システムである．

4.1 システム概要と設計

本研究の研究目的は，より簡単に複数人で協力的に多言語文書の同期を行うことである．SynCLDoc では多言語文書の各言語の版の内容の差分とその差分が他の言語の版のどこに対応するかをユーザーに示すことにより，比較的に同期しやすい環境を提供することを目指している．SynCLDoc は図 4.1 に示すように，ユーザーインターフェースとバックエンド二つの部分がある．

SynCLDoc はウェブアプリケーションであり，複数人で協力的に同期化を行うことが可能である．ユーザーはウェブブラウザにより多言語文書の作成，アップロード，編集と同期化作

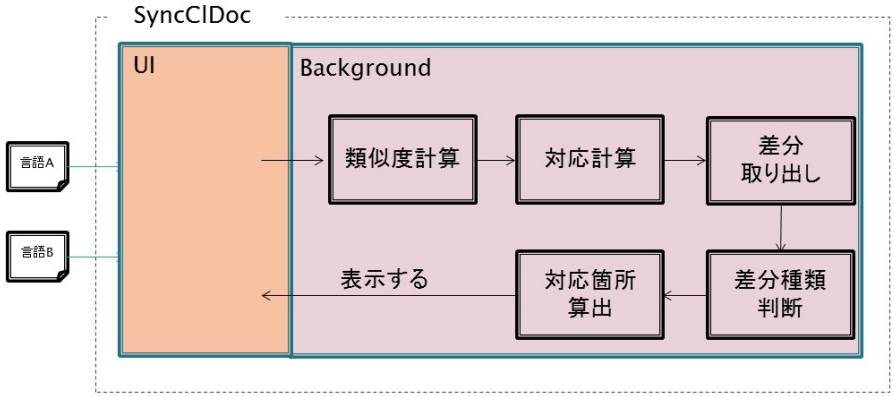

図 4.1. システム概要図

業を行う．SynCLDoc のバークエンドには既存の文の類似度計算アルゴリズムによる多言語文書の各言語の版の対応関係を計算し，差分を取り，そして差分の対応箇所を特定し，これらの情報をユーザーに示す．

4.2 ユーザーインタフェースと利用手順

複数の人で多言語文書の同期化が行えるため，SynCLDoc をウェブアプリケーションとして開発し，コモディティ化したウェブブラウザで利用できるようにした．また，ユーザーグループに分けて，多言語文書の作成や編集などが簡単にできるユーザーインターフェースを設計した．そして，多言語文書の各言語の版の独立した編集や，同期化を簡単にするため，それぞれの編集画面を用意した．各言語の版の古いバージョンの閲覧，各言語の版の文書の状態などのチェックも画面上で簡単にできる．さらに，各言語の版の対応関係を正確に取るために，対応関係の確認，修正画面を用意し，ユーザーが操作しやすいユーザーインターフェースを実装した．

本稿では，一つの具体的な例を挙げて（シーン 1 からシーン 7 の操作手順），SynCLDoc の利用手順を示す．この例には，3 人のユーザーがいており（田中さん，John，李さん），彼らは多言語文書 D を協力的にメンテナンスする．多言語文書 D は日本語版，英語版，中国語版の三つの言語の版がある．田中さんが日本人で，彼は日本語と英語ができる．John はアメリカ人で，彼は英語と中国語が話せる．そして李さんは中国人で，中国語と日本語ができる．

田中さん，John，李さんは多言語文書 D の各言語の版を修正することができる．もちろん，一つの言語の版だけに修正を与えることも許す．彼らは多言語文書 D の各言語の版の内容が同じであることを目標する．本稿では，多言語文書 D の作成から，編集や，同期化などの作業による各言語の版の状態を説明する．また，SynCLDoc はどのように複数のユーザーで多言語文書の同期化を行うことを支援するかを示す．

■シーン 1　多言語文書 D は日本語版，英語版，中国語版の三つの版が既に存在する．図 4.2 に示すように，英語版と中国語版には三つの段落があるが，日本語版に二つの段落（英語版の最初の段落と最後の段落）しかない．ただし，段落の順番は同じである．また，英語版の最初の段落の二文目，日本語版の二段落目の最後の文はそれぞれ他の二言語の版に存在しない．

■シーン 2　田中さんは SynCLDoc にログインし，ファイル一覧画面（図 4.3）で英語版をアップロードする．SynCLDoc のファイル一覧画面は SynCLDoc に存在するファイルの一覧画面である．この画面で，新しい多言語文書の作成やアップロードもできる．最初には，多言語文書 D はまだ SynCLDoc に存在しないため，ファイル一覧画面からアップロードする必要がある．

田中さんは多言語文書 D の英語版をアップロードした後，英語版を SynCLDoc にアップロードする．その場合，SynCLDoc に既に多言語文書 D が存在するため，多言語文書 D のある言語の版を選択し，編集画面を開いで「Upload」機能を利用し，他の言語の版をアップロードまたは作成する．実験には，田中さんが英語版の編集画面を開い，関連の日本語版をアップ

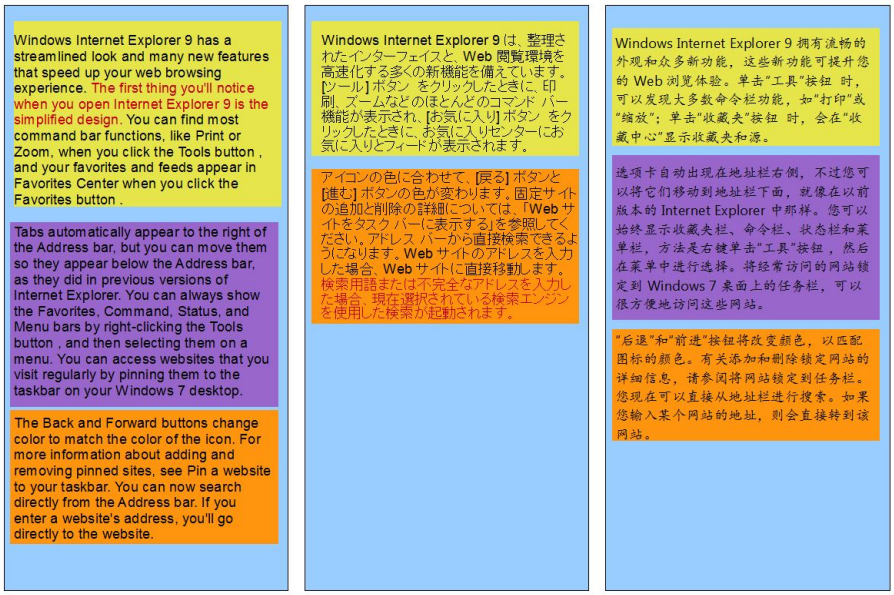

図 4.2. SynCLDoc ファイル一覧画面

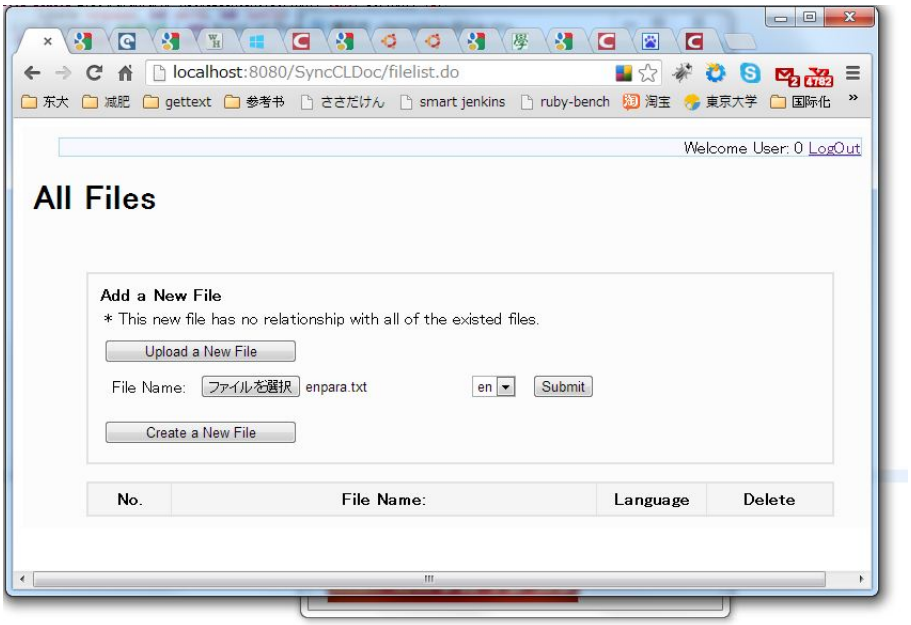

図 4.3. SynCLDoc ファイル一覧画面

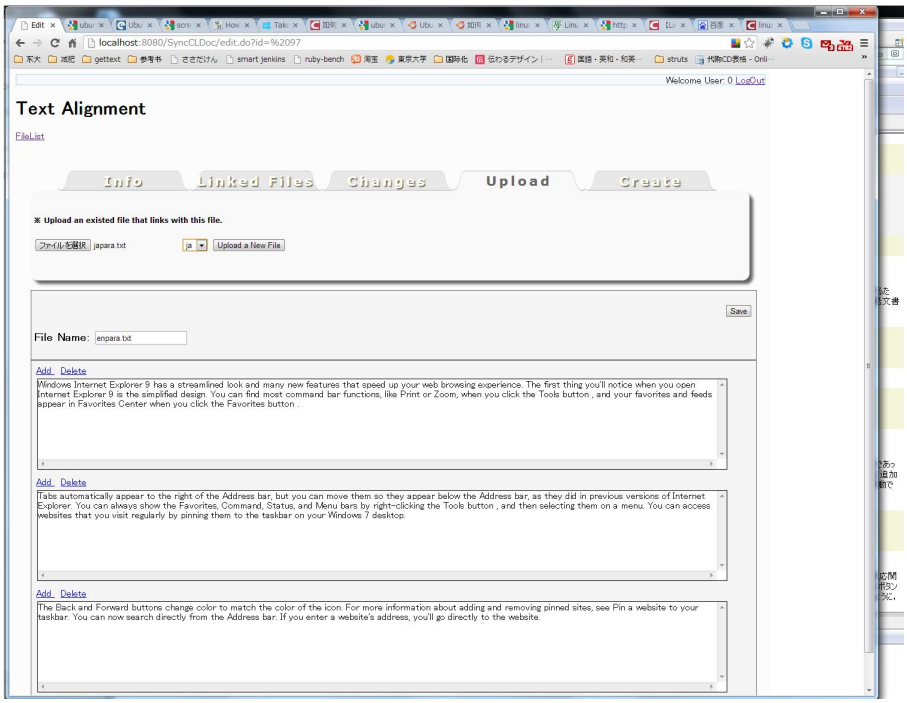

図 4.4. SynCLDoc に存在する多言語文書の他の言語の版をアップロードする画面

ロードする（図 4.4）.

アップロードを完了する際に，SynCLDoc は自動的に英語版と日本語版の内容の対応関係を計算し，差分とその対応箇所をユーザーに示す（図 4.5）．SynCLDoc が計算した対応関係は完全に正確ではないため，ユーザーは段落と文の対応関係の確認と修正作業を行う必要がある．初めに多言語文書の二つの言語の版を対応付ける時，対応関係の確認作業が必要である（図 4.5 の画面上部の赤い文字でメッセージを表示する）．段落のラジオボタンで指定することで，段落の対応関係の修正を行うことができる．各段落のチェックボックスにより，特殊段落の指定もできる．一度段落の対応関係を修正したら，画面上部「Save Aligned Relationship」ボタンを押すと，システムは自動的に段落内の文の対応関係を再計算する．また，各段落内の文の対応関係の確認と修正は各段落の「Confirm」 ボタンにより行う．「Confirm」ボタンを押すと，文の対応関係の確認と修正画面（図 4.6）が表示される．特殊段落の指定と同じように，特殊文の指定も各文のチェックボックスで指定する．

図 4.5 に示すように，SynCLDoc の計算により，英語版の一段落目は日本語版の一段落目，二段落目は日本語版の二段落目に対応している．しかし，この対応関係には間違うところがあるため，田中さんは手作業で段落の対応関係を修正する必要がある．順番として、まず英語版と日本語版の段落の対応関係を調整し（英語版の第三段落は日本語版の第二段落と対応する），段落の対応関係を修正した．そして，英語版の第一段落と日本語版の第一段落の対応関係が正しいが，文の対応関係には間違うところがある（実際には，英語版の二文目は差分である）．従って，第一段落の文の対応関係の修正も必要である（図 4.5）.

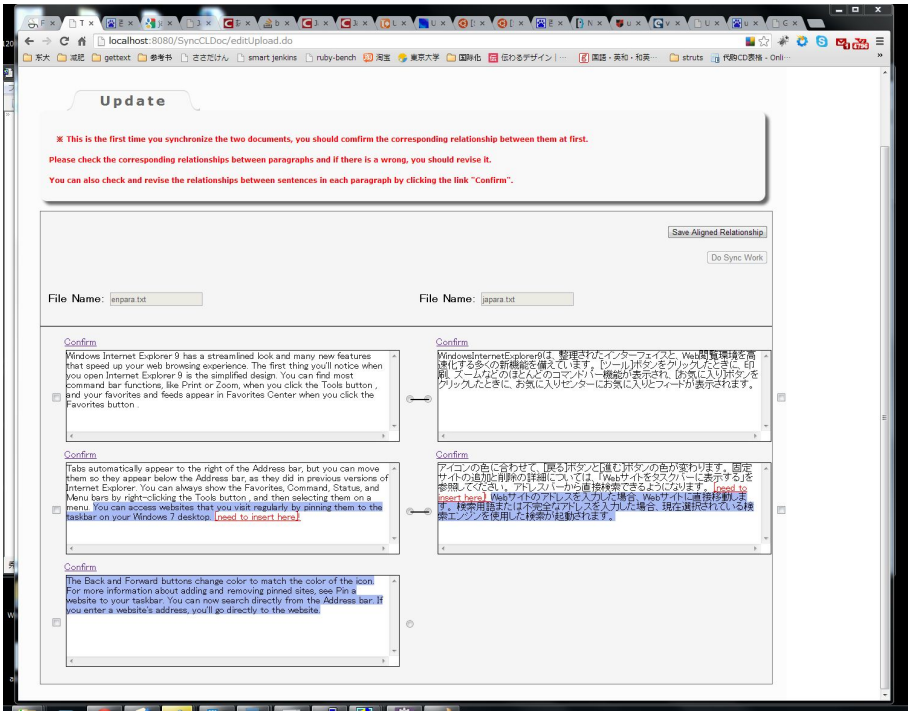

図 4.5. SynCLDoc の段落の対応関係の確認と修正画面

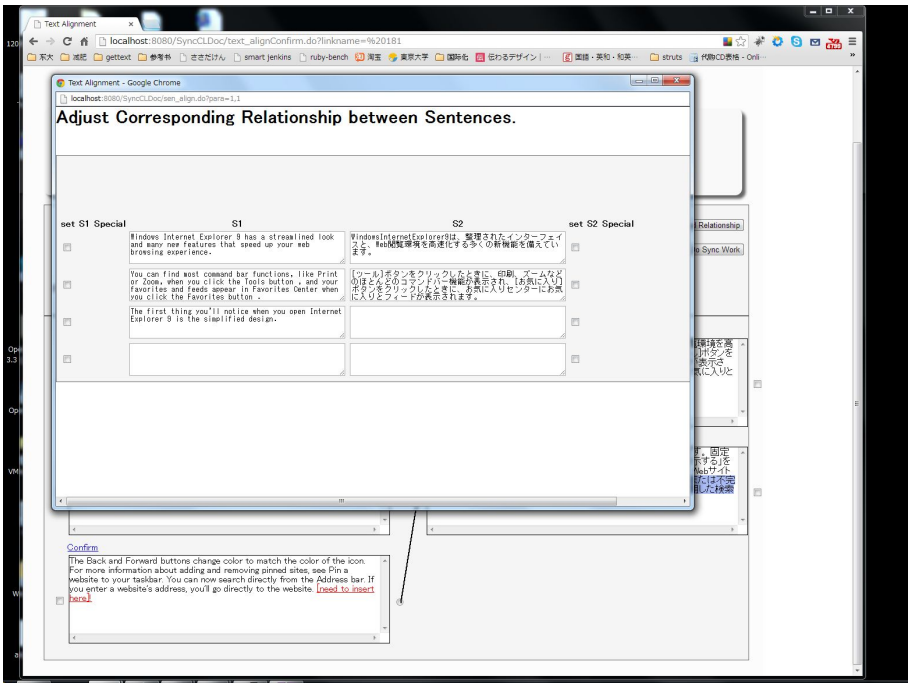

図 4.6. SynCLDoc の文の対応関係の確認と修正画面

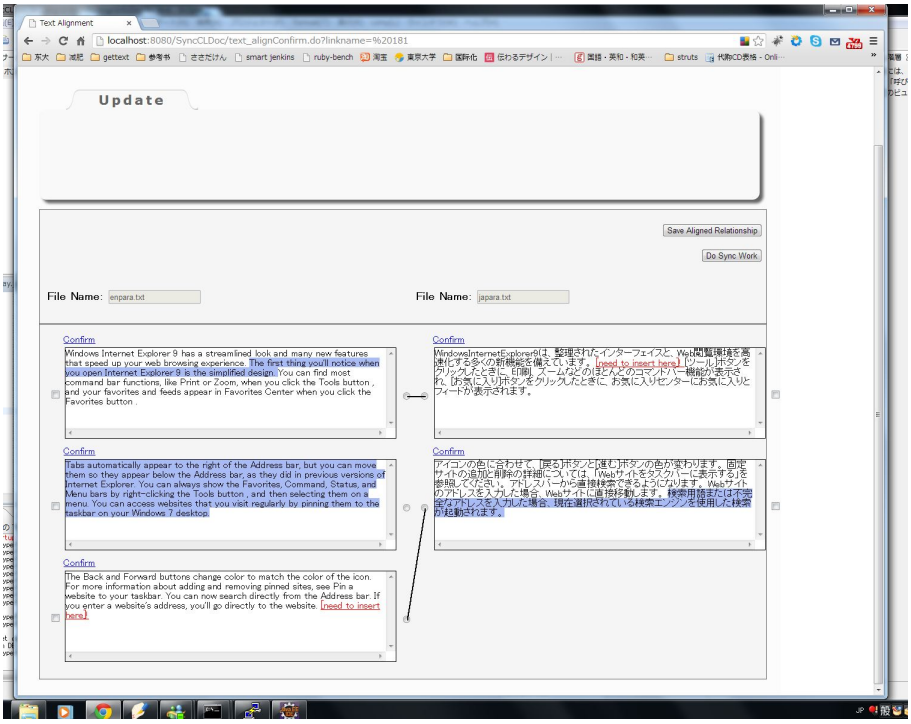

図 4.7. SynCLDoc の文の対応関係の確認と修正画面

多言語文書の二つの言語の版の対応関係の確認と修正作業を行う度に，SynCLDoc は確認された対応関係により，再び二つの言語の版の差分を判定し，表示する．田中さんが対応関係の修正を終わったら，図 4.7 に示すように，SynCLDoc は多言語文書 D の英語版と日本語版の対応関係と差分とその対応箇所が正確に表示する．また，ユーザーが対応関係の確認作業を行ったため，画面上部の赤い文字で表示した確認が必要というメッセージが表示しなくなる．

■シーン 3　田中さんは英語版と日本語版の対応関係の確認をしたが，同期化作業を行わなく，ファイル一覧画面に戻る．翌日，John は同じ操作で中国語版を SynCLDoc システムにアップロードし，英語版と中国語版の対応関係を確認する．その時，英語版，日本語版と中国語版の内容はそれぞれ修正が必要なので，SynCLDoc はこの三つの版に「Old マーク」を付いて表示する（図 4.8）．

■シーン 4　田中さんは SynCLDoc の差分表示を参照し，同期画面で英語版と日本語版の同期化をした後に，ファイル一覧画面に戻る．図 4.9 に示すように，英語版と日本語版の内容は一致し，中国語版の内容は英語版と日本語版には全部存在するため，英語版と日本語版の「Old マーク」が消える．ただし，中国語版にはまだ差分が存在するため，「Old マーク」をそのまま表示する．

その時，田中さんが英語版の日本語版の同期化画面を開くと，図 4.10 に示すように，差分は全部なくなる．

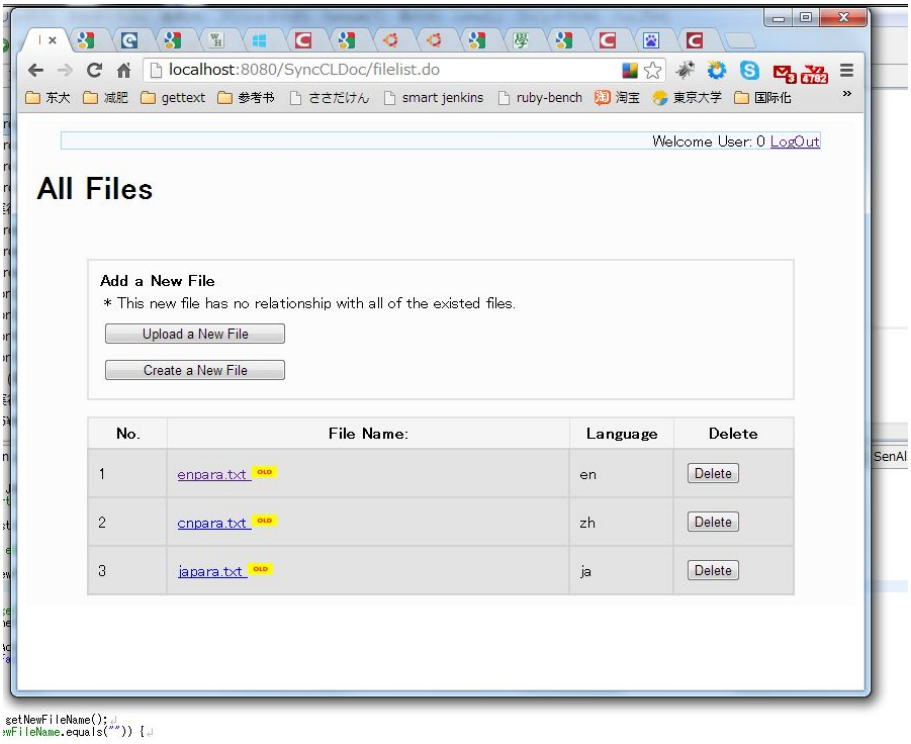

図 4.8. SynCLDoc ファイル一覧画面

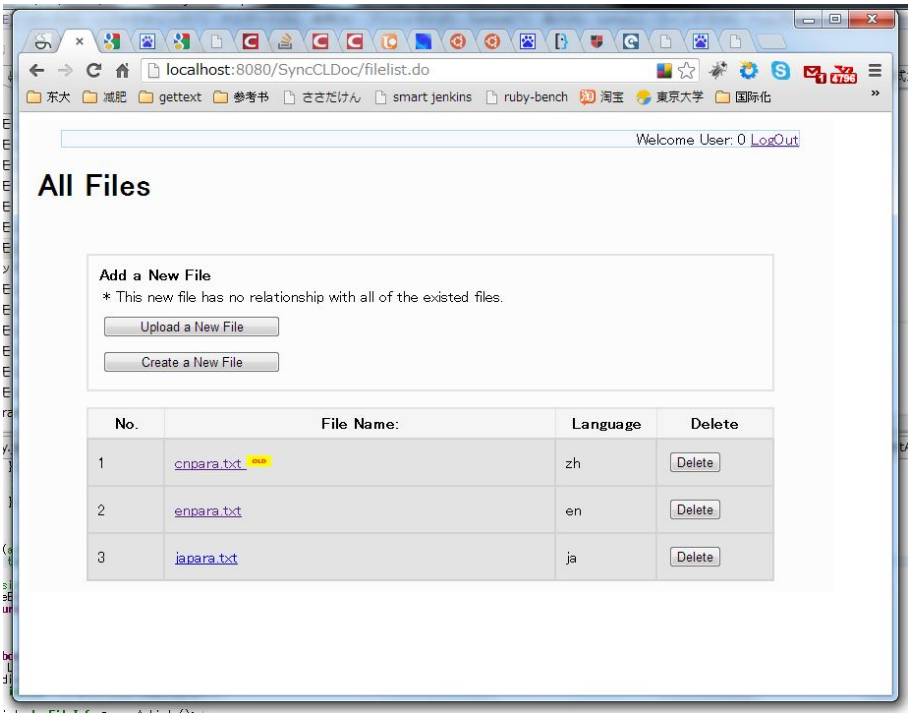

図 4.9. SynCLDoc ファイル一覧画面

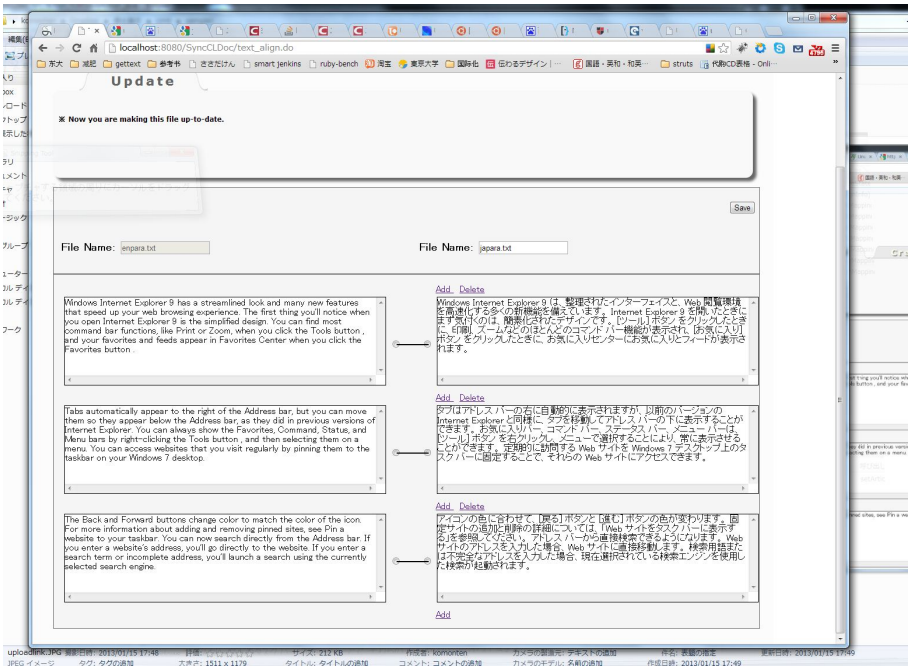

図 4.10. SynCLDoc 同期化画面

■シーン 5 翌々日に，李さんは SynCLDoc を利用し，日本語版と中国版の同期化を行う．その時，日本語版と中国語版との対応関係の確認が一度も行っていないため，まず対応関係の確認を行わなければならなかい．ただし，システムは自動的に日本語版と英語版の対応関係，英語版と中国語版の対応関係で，日本語版と中国語版の対応関係を再計算するため，日中の対応関係の誤差が少ない．ユーザーは日本語版と中国語版との対応関係は既存の日英と英中の対応関係から判定されたことが分かり（図 4.11 の上部で青い文字でメッセージを表示する），対応関係の確認と修正作業が手軽く行うことができる．

李さんは対応関係を確認した後，差分表示を参照し，同期画面で中国語版と日本語版の同期化をう．そうすると，三つの言語の版の状態は全部同じになる．

■シーン 6 そして，John は SynCLDoc を利用して，編集画面（図 4.12）で英語版を編集する．この時点で，英語版の内容が更新されたため，画面上部に「This file is up-to-date.」というメッセージを表示する．John は，一段落目の二文目を「The first thing you'll notice when you open Internet Explorer 9 is the simplified design. 」から「The first thing you'll notice when you open IE9 is its design. 」 に変更し，二段落目の第一文「」 を削除し，最後の段落の最初に文「Click the address bar to select your search engine from the listed icons or to add new ones.」 を追加する三つの修正を英語版に与える．

すると，日本語版と中国語版の同期化作業が必要になる．SynCLDoc のファイル一覧画面に，日本語版と中国語版に「Old マーク」が付く．

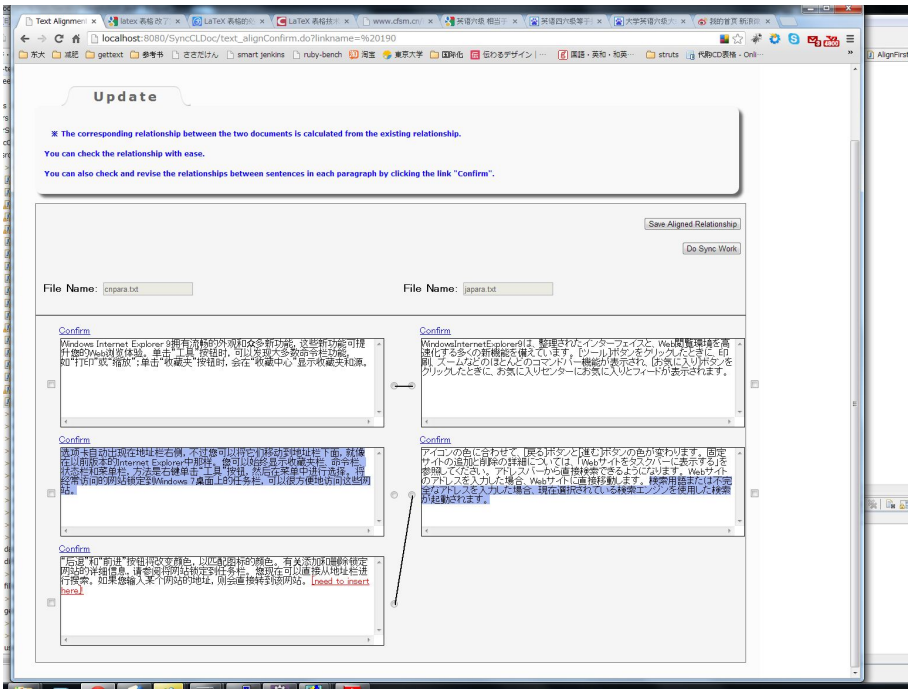

図 4.11. SynCLDoc の段落の対応関係の確認と修正画面:日英と英中の対応関係から日中の対応関係を判定する

図 4.12. SynCLDoc 編集画面

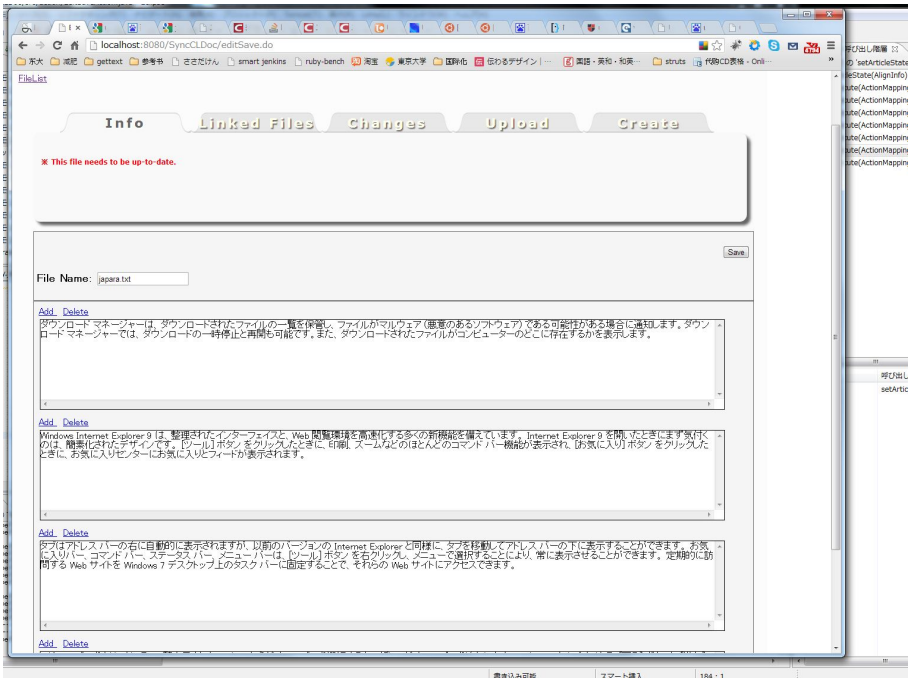

図 4.13. SynCLDoc 編集画面

■シーン 7　田中さんは John の修正の後に，SynCLDoc を利用して，編集画面（図 4.13）で日本語版を編集する．その時，日本語版を同期化する必要があるため，SynCLDoc は編集画面の上部に（図 4.13）示すように，赤い文字で「This file needs to be up-to-date.」というメッセージが表示する．田中さんは同期化作業を行わず，文書の最初に新しい段落を追加し，最後の段落の最初に文「アドレス バーをクリックして，一覧表示されたアイコンから検索エンジンを選択するか，新しい検索エンジンを追加します.」 を追加する二つの修正を日本語版に与える．

John と田中さんは二人とも最後の段落の初めに一つの文を追加したが，この二つの文が同じ意味のため，日本語版と英語版にこの修正の同期作業の必要がない．SynCLDoc は編集された文の類似度を計算し，編集箇所等が同じかどうかを判断するため，差分の取得がより正確である．SynCLDoc は追加された文を青い背景色で表示し，削除された文を赤い背景色と削除線で表示し，変更された文を黄色背景色で表示する．また，マウスを変更された文に移動したら，変更前の文の内容を表示する．対応編集の必要がある文を赤い文字とアンドラインで表示する．図 4.14 に示すように，英語版には，日本語版の最初の段落が存在しないため，追加する必要がある．また，英語版の二段落目の削除された文は日本語の二段落目の一文目と対応していたので，日本語版の二段落目の一文目を削除する必要がある．ただし，二つの言語の版の最後の段落に追加された文が同じ意味のため，追加する必要はない（画面上に差分ではないと表示している）．

本稿は上記の例を通して，SynCLDoc の利用手順を紹介した．SynCLDoc は多言語文書の

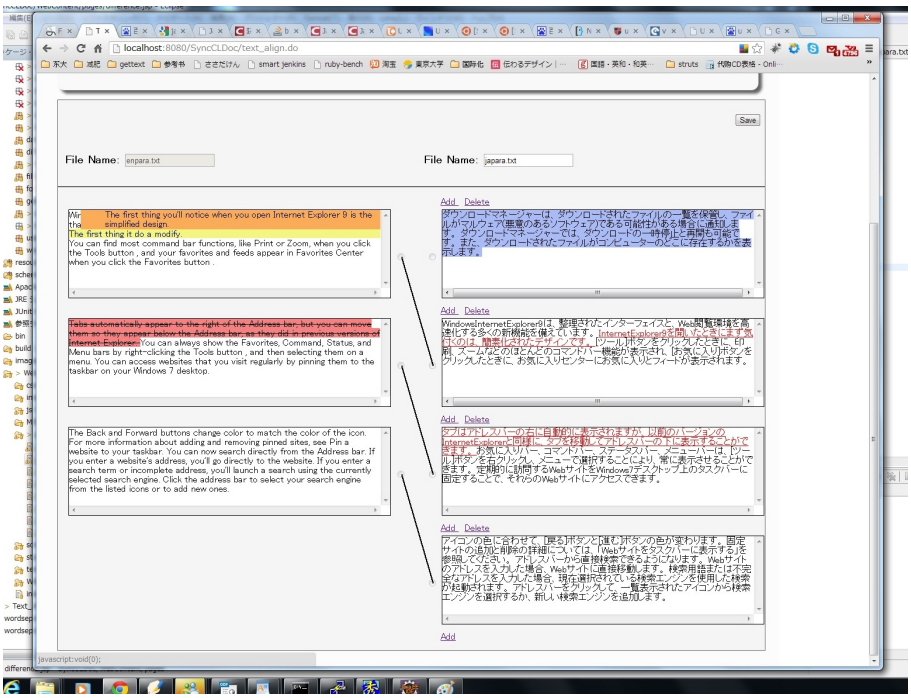

図 4.14. SynCLDoc 編集画面

段落と文の対応関係により，異なる言語の版の差分とその差分が対応する修正箇所をユーザーに分かるように表示し，多言語文書の同期しやすい環境を提供し，同期化を支援することが可能である．

4.3　修正箇所の特定の実装

本研究は，任意の文の類似度計算アルゴリズムを利用し，多言語文書の各言語の版の対応関係を計算することで，多言語文書の差分を取り出し，そして，差分の対応箇所を特定し，ユーザーに表示することにより，多言語文書の同期化を支援することを提案した．

本節では, まずいくつかの既存の文の類似度計算アルゴリズムを紹介する．次に, SynCLDoc システムに投入した文の類似度アルゴリズムについて詳しく説明する．そして，この文の類似度アルゴリズムをどのように利用し，多言語文書の対応関係を計算するかを述べる．また，文の類似度計算アルゴリズムによる計算された対応関係の誤差を減らすため，他の言語ペアを活用した新たな対応関係の判定の実装について説明する．さらに，多言語文書の各言語の版に存在する特殊段落と特殊文の対応関係と差分の取り扱い方法について述べる．最後は，システムによる多言語文書の間違い対応関係の修正方法について説明する．

4.3.1 類似度アルゴリズムを利用した二つの言語の版の対応関係の計算の実装

多言語文書の同期化する過程には，各言語の版の内容を比較し，差分を取り出して，修正する必要がある．システムによる各言語間の内容を比較し，一致しない内容をユーザーに示したら，手作業での差分を取り出す手間を減らすことができ，多言語文書の同期化を効率的にできるようになると考えられる．

本研究は，文の類似度計算アルゴリズムで各言語の版の内容が一致するかどうかを判断できると考えた．そこで，文の類似度計算アルゴリズムで多言語文書の対応関係を計算する方法を提案した．

類似度計算アルゴリズムについて

自然言語処理分野で，文の類似度計算アルゴリズムが多数存在する．

その中の一つは，文の長さによる類似度を計算する [19] [20]．これらの類似度計算方法は西洋語同士の類似度計算によく利用され，計算速度が速く，正確率も良い。しかしながら，日本語や中国語などの場合は，正確率は低い．例としては，a) 英語， b) フランス語， c) ポルトガル語， d) 日本語，e) 中国語の五種類の言語がある場合，

a. I want to play a game.

b. Je veux jouer a un jeu.

c. Eu quero jogar um jogo.

d. ゲームをやりたい.

e. 我想玩一場遊戲.

英語，フランス語，ポルトガル語の長さが同じであり，しかも全部スペースで単語を区切ることができるが，日本語と中国語の文の長さが定義しづらい．例えば，e) 文の中国語には「一場」は「一」と「場」二つの単語があるが，e) 文に「一場」は英語の「a」 と同じ意味である．また，d) 文の日本語には，英語の「a」と同じ意味の単語は存在しない．

または，機械翻訳などを利用し，文の意味で類似度を計算する [21, 22, 5, 23, 24]．これらの類似度計算方法は対訳データベースが必要とし，計算速度も長さによる方法より遅い．また，計算された類似度は対訳データベースの良さを大きく影響が与えられるが，対訳データベースがあるさえ，計算された類似度の有効性が比較的に高い．

利用した類似度計算アルゴリズム

本研究は [5] に提案された文の類似度計算アルゴリズム $SIM(S1, S2)$ を利用した．

$SIM(S1, S2) = \frac{co(S1 \cap S2)+1}{|S1|+|S2|-2co(S1 \cap S2)+2}$

$S1$ は言語 1 の単語の集合である．$S2$ は言語 2 の単語の集合である．$f(x)$ は単語 x が集合 X に出る頻度で，$|X| = \sum_{x \in X} f(x)$．$co(S1 \cap S2) = \sum_{(s1,s2) \in S1 \cap S2} min(f(s1), f(s2))$．$S1 \cap S2$ はある $(s1, s2) s1 \cap S1 \wedge s2 \cap S2$ に対して，$s1$ の訳文は $S2$ にあり，そして，$s2$ の

訳文は $S1$ に存在することである．

$SIM(S1, S2)$ の評価結果によると，類似度が 30% 以下の場合，二つの文が同じではないと判断される．類似度が 30% 以上が，50% 以下の場合，二つの文は大体同じ意味であると判断される．類似度が 50% 以上の場合，二つの」文は同じ文であると判断される．. これにより，1989 年から 2001 年の約 944,404 篇の日本語と英語記事の対応付けの正確率は 95% である．

対応関係の計算アルゴリズム

本研究は Java でウェブアプリケーション SynCLDoc を実装した．SynCLDoc は日本語，英語と中国語を支援しており，$SIM(S1, S2)$ を利用するため，日本語，英語，中国語をそれぞれの文書から単語を取り出す必要がある．また，各言語の同義語と他の二つの言語の訳語が必要である．

各言語の単語の取得は既存のツールやライブラリーで実現した．日本語文書から日本語の単語を取り出すに関する日本語形態素分析の研究が多数存在する [25, 26, 27]．また，中国語文書から中国語の単語を取り出すに関する中国語形態素分析の研究もいくつか存在する [28, 29, 30] ．本研究は Kuromoji[*1] という日本語形態素解析ツールを利用し，日本語文書の単語を取得する．IK Analyzer [*2] を利用し，中国語の単語を取得する．また，WordNet[*3] の英語版と日本語版を利用し，英語と日本語の連語と複合語を取得し，同義語と訳語も取得する．WordNet が提供している対訳語の数が足りないため，本研究は最初に Microsoft Translator [*4]で単語の同義語と訳語を取得することを実装したが，Microsoft Translator はネットサービスなので，大量の単語の同義語と訳語を取得することに大量の時間がかかるため，システムに投入しなかった．対訳データベースの不足を解決するため，本研究は Lingoes [*5] が提供している辞書を利用し，英日，英中，日中の対訳と各言語の同義語のデータベースを作成した．また，Redis [*6] というオープンソースの key-value データベースを利用した．

本研究は拡張しやすい多言語文書の対応関係のデータ構造を設計し，実装した．一つの多言語文書は一つの対応関係を持ち，多言語文書の各言語の版の内容が修正されたら，対応関係を修正する必要がある．多言語文書 D の対応関係 R は複数の文の対応関係 Rs を持っている．Rs は複数の情報 [言語，文書 ID，段落 ID, 文 ID，修正フラグ，特殊文フラグなど] を持っている．各言語の版の段落や文が対応することを判定した際に，これらの段落や文の情報が一つの Rs に追加する．これにより，システムに新しい言語の支援を増加する際に，対応関係の構造の変更がしなくても，使用可能である．類似度計算アルゴリズムにより，言語 1 で書かれた段落と言語 2 で書かれた段落の対応関係を付ける場合，下記の二つの状況を分け，対応関係を追加する．

[*1] http://www.atilika.org/
[*2] http://code.google.com/p/ik-analyzer/
[*3] http://wordnet.princeton.edu/
[*4] http://www.microsoft.com/en-us/translator/
[*5] http://www.lingoes.cn/zh/index.html
[*6] http://redis.io/

- 段落情報が多言語文書の対応関係に既に存在しない場合，つまりこの二つの段落が両方とも他の言語の段落と対応付ていない場合
  言語 1 の段落の情報（文書 ID と段落 ID）と言語 2 の段落の情報を多言語文書の対応関係に追加する．
- 段落情報が多言語文書の対応関係に既に存在する場合，つまりこの二つの段落の一つまたは両方とも他の言語の段落と対応付けた場合
  既に存在する対応関係を取り出し，この対応関係に存在しない言語の段落の情報を追加する．例えば，言語 1 の段落 1 は言語 3 の段落 1 と対応付けている．この対応関係が多言語文書の対応関係に既に存在した．そのとき，言語 1 の段落 1 と言語 2 の段落 1 を対応付ける場合，言語 1 の段落 1 の情報が含めている対応関係に言語 2 の段落 1 の情報を追加する．

類似度計算アルゴリズムにる各段落の文の対応関係の追加も上記の方法と同じように実現した．

4.3.2 他の言語ペアを活用した新たな対応関係の判定アルゴリズムの実装と差分の表示

節 3.2 で提案した既存の言語ペア間対応関係から新しい言語ペア間の対応関係の計算は節 4.3.1 から計算された多言語文書の対応関係を分析し，言語間の対応関係を取り出すことで実現できる．

例えば，多言語文書 D が日本語版，英語版，中国語版が存在する．システムは日本語版と英語版，英語版と中国語版の対応関係を類似度計算による取得した．ユーザーはこの二つの対応関係を修正し，確認した．その場合，各文の対応関係 Rs には [日本語，日本語文書 ID，日本語段落 ID, 日本語文 ID，修正フラグ，特殊文フラグ]，[英語，英語文書 ID，英語段落 ID, 英語文 ID，修正フラグ，特殊文フラグ]，[中国語，中国語文書 ID，中国語段落 ID, 中国語文 ID，修正フラグ，特殊文フラグ] が存在する．そしたら，日本語版と中国語版の対応関係は下記のステップで簡単に取得できる．

1. 多言語文書 D の各文の対応関係 Rs から，Rs_je と RS_ec を取り出し，二つの言語の版の対応関係 Rs_jc を作成する．
2. 日本語版に存在する段落や文が Rs_jc に存在しない場合，段落と文を対応付け待ちリスト Ls_j に追加する．
3. 同様に，中国語版に存在する段落や文が Rs_jc に存在しない場合，段落と文を対応付け待ちリスト Ls_c に追加する．
4. 類似度計算アルゴリズムを利用し，Ls_j と Ls_c に存在する段落や文の対応関係を計算し，Rs_jc に追加する．

また，多言語文書の各言語の版の差分の判断と表示も計算された多言語文書の対応関係を分

析することにより行う．多言語文書の各文の対応関係 Rs に修正フラグがある．修正フラグは同様，古い二つの種類がある．初めに各言語の版の対応関係を付ける時，各文の対応関係 Rs に修正フラグを「同様」と設定する．対応付けた段落や文に一つの修正を与える場合，修正された文と対応している文（他の言語）の修正フラグを「古い」に設定する．複数の修正を各言語の版の対応付けた段落や文に与える場合，修正内容が同じだと判断したら，これらの言語の版の修正された文の修正フラグを「同様」に設定し，他の言語の版の対応文の修正フラグを「古い」に設定する．

SynCLDoc は対応関係に存在しない段落や文を差分として，また，対応関係に存在し，修正フラグが「古い」の段落や文も差分として，ユーザーに表示する．ユーザーが同期化作業を行う時，SynCLDoc は自動にユーザーが編集した内容が同期化内容なのか，新しい修正なのかを判定する．編集された内容は同期化内容の場合，二つの言語の版の修正フラグが「同様」に設定する．編集された内容は新しい修正の場合，上記の方法で修正フラグを設定する．

4.3.3　差分の修正方法と特殊文の処理

自動的に計算された対応関係による取得した差分とその差分が他の言語の版の対応箇所の表示が間違うこともある．そのため，SynCLDoc はユーザーが手作業で段落と文の対応関係の修正と確認ができるようにした．図 4.5 は段落の対応関係の修正と確認画面であり，段落の対応と特殊段落の指定ができる．また，図 4.6 は文の対応関係の修正と確認画面であり，文の対応関係の指定と特殊文の指定ができる．文の対応関係が存在しないの場合，対応する枠を空にすれば指定できる．

SynCLDoc は文化の理由で，特殊段落と特殊文の存在が許す．特殊段落と特殊文とは，多言語文書のある言語の版だけに存在する段落や文，または一部分の言語の版だけに存在する段落や文である．特殊段落と特殊文の指定はユーザーの手間がかかり，システムが自動的に指定しない．特殊段落と特殊文が存在する言語の版は対応関係による状態の変更や対応箇所の特定を行うが，特殊段落と特殊文が存在しない言語の版には影響を与えない．

第 5 章

評価

本章では，提案された類似度アルゴリズムを利用した二言語の対応関係の計算アルゴリズムと，他の言語ペアを活用した新たな対応関係の判定アルゴリズムに対する評価を述べる．また，開発した SynCLDoc システムのユーザー実験を行い，評価する．そして，これらの評価結果を踏まえた上で，本研究の提案手法の有効性について検証し，SynCLDoc はどういう条件で多言語文書の同期化を効率的に行えるかについて分析し，システムの有用性を示す．

5.1 類似度アルゴリズムを利用した二つの言語の版の対応関係の計算の評価

本節では，[5] に提案された文の類似度計算アルゴリズム $SIM(S1, S2)$ （節 4.3.1）を利用し，多言語文書の二つの言語の版の対応関係を計算するアルゴリズムの正確性を評価する．実験では，多言語文書の日本語版，英語版，中国語版の対応関係の計算を行い，その正確さを測定した．

実験に利用した対訳データベースは WordNet の英語バージョンと日本語バージョン，そして我々が作成した日中英の対訳データベースである．WordNet の英語バージョンの単語数は約 155,287 であり，日本語バージョンの単語数は約 93,834 である．我々が作成した英日の対訳データベースの単語数は 145,469 であり，英中の対訳データベースの単語数は 478,093 であり，日中の対訳データベースの単語数は 494,778 である．

本研究は計算された段落対応関係の正確さに段落スコア（0 点〜10 点），文の対応関係の正確さに文スコア（0 点〜10 点）で評価した．段落スコアと文スコアは下記のように定義される．

多言語文書 D の言語 X と言語 Y の段落スコア $Score(P_xy)$ は SynCLDoc が計算した言語 X の版と言語 Y の版の段落の対応関係の正確率を表すものである．SynCLDoc による計算された段落の対応関係が完全一致の場合，$Score(P_xy) = 10$ であり，完全不一致の場合，$Score(P_xy) = 0$ である．

$Score(P_xy) = \frac{Co(P_x + Co(P_y)}{|P_x| + |P_y|}$ $(0 \leq Score(P_xy) \leq 10)$

P_x は言語 X の版の段落の集合であり，$|P_x|$ は言語 X の版の段落の数である．P_y は言語 Y

の版の段落の集合であり，$|P_y|$ は言語 Y の版の段落の数である．$Co(P_x)$ は言語 X の段落に正確な対応関係を付けた段落の数である．$Co(P_y)$ は言語 Y の段落に正確な対応関係を付けた段落の数である．ただし，本研究は段落 px_i と段落 py_j,py_j+1,...$py_j+n(1 \leq j < j+n \leq |P_y|)$ が対応しているが，SynCLDoc は段落 px_i と段落 $py_k(j \leq k \leq j+n)$ しか判断できない場合，段落 py_k の対応関係が正確と定義し，他の段落の対応関係が正確ではないと定義する．

同じように，多言語文書 D の言語 X と言語 Y の文スコア $Score(S_xy)$ は SynCLDoc が計算した言語 X の版と言語 Y の版の文の対応関係の正確率を表すものである．SynCLDoc による計算された文の対応関係が完全一致の場合，$Score(S_xy) = 10$ であり，完全不一致の場合，$Score(S_xy) = 0$ である．

$Score(S_xy) = \frac{Co(S_x+Co(S_y)}{|S_x|+|S_y|}$ $(0 \leq Score(S_xy) \leq 10)$

S_x は言語 X の版の文の集合であり，$|S_x|$ は言語 X の版の文の数である．S_y は言語 Y の版の文の集合であり，$|S_y|$ は言語 Y の版の文の数である．$Co(S_x)$ は言語 X の文に正確な対応関係を付けた文の数である．$Co(S_y)$ は言語 Y の文に正確な対応関係を付けた文の数である．ただし，本研究は文 sx_i と複数の文 $Multi(sy), Multi(sy_n - j) = sy_j, sy_j + 1, ..., sy_j + n(1 \leq j < j + n \leq |S_y|)$ が対応しているが，SynCLDoc は段落 px_i と複数の文 $Multi(sy_m - j), Multi(sy_m - j) = sy_k, sy_k + 1, ..., sy_k + m(j \leq k < k + m < j + n)$ しか判断できない場合，文 $sy_k(\forall sy_k \in Multi(sy_m - j))$ の対応関係が正確と定義し，他の文の対応関係が正確ではないと定義する．

本研究は 100 文からなる日本語，英語，中国語の 15 篇の多言語文書の対応付けの正確率を測定した．その中，5 つの多言語文書はソフトウェアのマニュアル[*1]であり，5 つの多言語文書は小説[*2]であり，残った 5 つの多言語文書はブログや Wikipedia の記事や日記のような文書である．この 15 篇の文書の各言語の版の内容は完全一致のものがあり，部分一致のものがあり，まったく一致しないものもある．また，段落と文の順番が一致しない状況や，内容が重複でる状況も存在する．そして，多言語文書の各言語の版の段落と文が一対一のみと一対多の状況も存在する．既存の類似度計算アルゴリズムから二つの言語の版の対応関係の計算アルゴリズムによる計算された対応関係の段落スコアと文のスコアは表 5.1，表 5.2，表 5.3 に示す．

表 5.1 の 100 文からなるのソフトウェアマニュアルの対応付け結果によると，段落が一対一の対応関係があるマニュアルの文の類似度計算アルゴリズムから二つの言語の版の対応関係の計算アルゴリズムによる計算された段落の対応関係の正確率が高い．また，アルゴリズムによる計算されたマニュアルのような多言語文書の文の対応関係の正確率も高く，平均約 91% の文に正確な対応関係を付けることができた．ただし，マニュアル 2 の計算された文の対応関係の正確率が比較的に低い結果が出た．マニュアル 2 の各言語の版の段落の対応関係が低くないが，文の対応関係が低い．それは，マニュアル 2 の各言語の版に，短い文が多数存在するためである．類似度計算アルゴリズムは短い文に対してはうまく働かない原因ある．

[*1] 本研究の評価に使われたマニュアルの内容は Microsoft Office 2010 製品ガイドと Windows Internet Explorer 9 の使用ガイドから取り出したものである．

[*2] 本研究の評価に使われた小説の内容は「レ・ミゼラブル」と「吾輩は猫である」の日本語版，英語版と中国版から取り出したものである．

表 5.1. 100 文のソフトウェアマニュアルの対応付け結果

文書	言語ペア	段落スコア：一対一対応のみ	段落スコア：段落と文が一対多対応あり	文スコア：一対一対応のみ（段落が一対一のみ）	文スコア：文が一対多対応あり（段落が一対一のみ）
マニュアル 1	英日	10.0	7.5	9.5	9.7
	英中	9.1	7.2	9.4	9.7
	日中	9.1	7.2	7.6	7.5
マニュアル 2	英日	10.0	8.3	7.3	7.6
	英中	10.0	8.2	8.8	9.2
	日中	10.0	8.0	7.1	6.9
マニュアル 3	英日	10.0	6.5	10.0	10.0
	英中	10.0	6.5	9.1	9.4
	日中	10.0	6.3	9.3	9.4
マニュアル 4	英日	9.5	4.2	9.6	9.7
	英中	9.5	4.2	8.6	9.0
	日中	9.5	3.8	8.9	8.7
マニュアル 5	英日	10.0	6.3	9.7	10.0
	英中	9.4	5.9	9.4	9.7
	日中	10.0	6.3	9.6	9.7

表 5.2. 100 文の小説の対応付け結果

文書	言語ペア	段落スコア：一対一対応のみ	段落スコア：段落と文が一対多対応あり	文スコア：一対一対応のみ（段落が一対一のみ）	文スコア：文が一対多対応あり（段落が一対一のみ）
小説 1	英日	10.0	6.4	3.7	3.7
	英中	9.2	5.8	4.1	4.3
	日中	7.0	5.3	1.8	2.1
小説 2	英日	9.2	7.8	3.6	3.4
	英中	7.0	5.3	2.9	2.8
	日中	7.7	5.9	3.3	3.3
小説 3	英日	10.0	5.5	2.6	2.7
	英中	6.7	3.6	3.2	3.0
	日中	9.0	5.2	3.3	3.8
小説 4	英日	6.2	4.2	4.5	3.0
	英中	5.8	4.5	4.2	3.7
	日中	5.3	3.8	3.9	3.5
小説 5	英日	4.5	2.3	3.6	2.2
	英中	4.1	1.7	3.8	2.3
	日中	3.4	1.6	2.7	1.1

また，表 5.2 と表 5.3 の結果によると，本研究が提案した既存の文の類似度計算アルゴリズムから二つの言語の版の対応関係の計算アルゴリズムにより，小説や記事などの多言語文書に対して，いい効果が得られなかった．それは，小説の各言語の版には，意訳の文が多く，文の類似度計算アルゴリズムによる二つの言語の文が同じかどうかを判定できないためである．記事などには，新語や固有名詞などの単語が多数存在し，既存の対訳データベースから単語の訳語や同義語などの抽出ができなく，計算された類似度が低くなるため，対応関係の計算もでき

表 5.3. 100 文の記事や日記のような文書の対応付け結果

文書	言語ペア	段落スコア：一対一対応のみ	段落スコア：段落が一対多対応あり	文スコア：一対一対応のみ（段落が一対一のみ）	文スコア：文が一対多対応あり（段落が一対一のみ）
記事や日記 1	英日	10.0	8.2	8.1	8.1
	英中	10.0	8.2	7.9	8.0
	日中	10.0	8.2	7.6	7.5
記事や日記 2	英日	8.6	8.3	7.4	8.5
	英中	8.6	8.3	7.2	8.6
	日中	8.6	8.3	6.8	7.3
記事や日記 3	英日	7.8	6.9	5.5	5.3
	英中	7.9	7.2	5.6	5.3
	日中	6.7	6.3	4.7	4.2
記事や日記 4	英日	8.9	5.4	6.2	8.7
	英中	8.9	5.4	6.4	8.3
	日中	8.6	4.9	5.6	8.0
記事や日記 5	英日	7.3	6.4	4.5	4.3
	英中	7.0	6.2	4.1	3.9
	日中	6.7	5.5	3.6	2.8

なくなる．

さらに，表 5.1，表 5.2 と表 5.3 に示すように，多言語文書の各言語の文が一対一に対応するか，一対多に対応するかは文の対応関係の計算に大きな影響を与えない．一方，段落が一対多の対応関係が存在する多言語文書の対応関係の正確率は低い．その原因は，本研究が提案した類似度計算による段落の対応関係を計算するアルゴリズムは一対多の段落の対応関係の計算を行わないためである．従って，マニュアルや，小説や，記事などの多言語文書に段落の対応関係が一対多の段落が多く存在するほど，計算された段落の対応関係のスコアが低くなる．

測定の結果によると，本研究が提案した多言語文書の二つの言語の版の対応関係を計算するアルゴリズムはマニュアルのような多言語文書の対応関係の計算は最適であることを確認した．小説のような文学性が強い文書や記事のような新語が多い文書などの多言語文書の対応関係の計算結果は良くないが，これらは本研究が提案したシステムの同期化支援する対象外ではない．

本研究が提案されたアルゴリズムは単純であるが，マニュアルのような多言語文書の対応関係の計算は効果的であることを確認できた．もっと良い対応関係の計算アルゴリズムを利用したら，計算された多言語文書の対応関係の正確率はより高くなると考えられる．

5.2　他の言語ペアを活用した新たな対応関係の判定の評価

本節では，本研究が提案した多言語文書の他の言語ペアを活用した新たな対応関係の判定アルゴリズムを利用して，多言語文書の対応付けた言語ペア間の対応関係から新しい言語ペア間の対応関係を計算するアルゴリズムの性能を評価する．実験では，多言語文書の日英の版の対応関係，日中の版の対応関係から英中の版の対応関係の計算，日英の版の対応関係，英中の版

表 5.4. マニュアルの他の言語ペアを活用した新たな対応関係の判定結果

文書	既存の対応関係	計算する対応関係	対応関係スコア（5篇のマニュアルの平均のスコア）
マニュアルセット 1	英日，英中	日中	10.0
マニュアルセット 1	英日，日中	英中	10.0
マニュアルセット 1	日中，英中	英日	10.0
マニュアルセット 2	英日，英中	日中	9.2
マニュアルセット 2	英日，日中	英中	9.7
マニュアルセット 2	日中，英中	英日	9.7
マニュアルセット 3	英日，英中	日中	8.2
マニュアルセット 3	英日，日中	英中	8.8
マニュアルセット 3	日中，英中	英日	8.9

の対応関係から日中の版の対応関係の計算と，日中の版の対応関係，中英の版の対応関係から英日の版の対応関係の計算の三つの計算を行い，その正確さを測定した．そして，計算された結果と節 4.3.1 に実装された類似度で二つの言語の版の対応関係の計算アルゴリズムの結果を比較する．

本実験は 3 つのマニュアルセットと 3 つの小説セットを利用して行った．各マニュアルセットと小説セットに 5 篇の多言語文書が存在する．また，各言語ペアの版の既存の対応関係が全部正確である．マニュアルセット 1 と小説セット 1 の各言語文書の各言語の版が完全一致している．つまり，各言語の版の段落と文が全部対応関係に存在することである．マニュアルセット 2 と小説セット 2，マニュアルセット 3 と小説セット 3 の各言語文書の各言語の版がそれぞれ違うところがある．マニュアルセット 2 と小説セット 2 の各言語の版が 80% ぐらいに他の言語の版に存在する．つまり，各言語の版の 20% ぐらいの文が既存の対応関係に存在しない．マニュアルセット 3 と小説セット 3 の各言語の版が 50% ぐらいに他の言語の版に存在する．つまり，各言語の版の 50% ぐらいの文が既存の対応関係に存在しない．

また，実験に使用したデータベースは節 5.1 の実験と同じである．実験結果は表 5.4，表 5.5 を示した（文スコアの計算方法は節 5.1 と同じである）．

表 5.4 と表 5.5 で示すように，他の言語ペアを活用した新たな対応関係の判定アルゴリズムの効果は多言語文書の各言語の版の文が既存の対応関係にどれくらいの割合で存在するかの影響を受ける．その原因は，新しい言語ペア間の対応関係を計算する時，新しい言語ペアの二つの言語の版の中に，対応関係に存在しない文は類似度計算アルゴリズムにようる対応関係を計算するためである．従って，各言語の版の文が全部既存の対応関係に存在する場合，新たな言語ペアの版の対応関係がほぼ 100% 正確であるが，各言語の版の文が既存の対応関係に存在し

表 5.5. 小説の他の言語ペアを活用した新たな対応関係の判定結果

文書	既存の対応関係	計算する対応関係	対応関係スコア（5篇の小説の平均のスコア）
小説セット 1	英日，英中	日中	10.0
小説セット 1	英日，日中	英中	10.0
小説セット 1	日中，英中	英日	10.0
小説セット 2	英日，英中	日中	7.8
小説セット 2	英日，日中	英中	8.4
小説セット 2	日中，英中	英日	8.2
小説セット 3	英日，英中	日中	5.3
小説セット 3	英日，日中	英中	5.1
小説セット 3	日中，英中	英日	5.2

ない文が多いほど，計算された結果は文の類似度計算アルゴリズムによる計算された対応関係と近い．

表 5.4 と表 5.5 によると，他の言語ペアを活用した新たな対応関係の判定アルゴリズムによる得られた対応関係は直接に類似度計算アルゴリズムを利用するより，よい結果が出る．そのため，他の言語ペアを活用した新たな対応関係の判定アルゴリズムによるユーザーの対応関係の確認の手間を減らすことができる．例えば，ユーザーは英語版と日本語版の対応関係を修正し，英語版と中国語版の対応関係を修正した場合，日本語版と中国語版の対応関係の正確率が二言語の対応関係から新しい対応関係を計算アルゴリズムにより高くなるため，ユーザーの修正と確認の時間が減らせるとこが確認できた．また，表 5.5 によると，一度対応つけた小説のような多言語文書でも，文の類似度計算アルゴリズムにより計算された対応関係の正確率が高いことも分かった．

5.3　システム実験による評価

本節では，ユーザーを実際に SynCLDoc を使い，多言語文書の同期化を行い，多言語文書の同期化時間を測定し，SynCLDoc の有用性を評価する．

ユーザー実験では，日本語版，中国語版と英語版がある 21 篇の多言語文書を利用して行った．その中，ソフトウェアマニュアルは 12 篇があり，下記には M1〜M12 で表示する．また，小説や普通の文書は 9 篇があり，下記には D1〜D9 で表示する．ユーザー実験に参加する人数は 18 人がいる．この 18 人が全部中国人であり，中の 10 人が日本語（日本語能力試験 N2

以上のレベル[*3]，英語（CET6 のレベル[*4]）と中国語（母語）ができ，8 人が英語（CET6 のレベル）と中国語（母語）ができる．各実験に 10 人が参加し，5 人が SynCLDoc を利用し，5 人が多言語文書の修正履歴だけを表示するシステム（下記は System B で表示）を利用し，実験 1〜実験 4 のユーザー実験を行った．

■実験１：対応つけていない二言語の既存文書の同期化：英日 既存の 5 つの多言語文書 M1-M3, D1-D2（英語版と日本語版がある）を利用して，英語版と日本語版の同期化実験を行った．日本語と英語が出来る 10 人を SynCLDoc を利用するチーム（5 人）と，System B を利用するチーム（5 人）に分けて，実験した．M1 と D1 の各言語の版は完全一致で（各言語の版が 50 文で 10 段落からなる），M2 と D2 の各言語の版にそれぞれ違うところがあり（各言語の版が 50 文で 11 段落からる．全部 10 ヶ所が違う），M3 の各言語の版の内容はまったく一致しない（各言語の版が 50 文からなる）．各システムの同期化時間はそれぞれのシステムを利用する 5 人が一つの多言語文書の差分を探し出す時間の平均値である．各システムの見逃し率はそれぞれのシステムを利用する 5 人が自分自身が全部の差分を取り出したと思ったが，文書にまた存在する差分数のパーセント（見逃した差分数/全部の差分数）の平均値である．

表 5.6 に示すように，SynCLDoc を利用した場合と System B を利用した場合，対応つけていない多言語文書の二言語の版を同期化する時間は大きな差が存在しなかったが，SynCLDoc を利用した場合，見逃し率が比較的に少ない．なぜなら，SynCLDoc を利用し，多言語文書を初めに同期化する場合，ユーザーは SynCLDoc が計算した対応関係を確認と修正が必要である．初めに対応関係の確認を行うとき，ユーザーは全部の段落と文の対応関係をチェックすることが必要である．それは System B を利用し，差分を取り出す作業と同じくらいの作業量のため，同じくらいの時間がかかる．ただし，ユーザーが SynCLDoc を利用する時，大体の対応関係を確認すると（例えば，段落の対応関係だけをチェックする），同期化時間を減らすことができる．

一方，SynCLDoc を利用し，対応関係を修正する作業も必要のため，SynCLDoc に詳しくないユーザーは逆に時間がかかる場合もある．SynCLDoc はユーザーに段落の対応関係と文の対応関係，計算した差分も目立ってユーザーに表示するため，差分が取り出されないことが避けることができた．また，小説のような多言語文書は意訳が多いため，平均の同期化時間がマニュアルより長かった．特に，SynCLDoc が小説のような多言語文書の対応関係の計算がうまくできなく，ユーザーの修正する手間もマニュアルよりかかった．

■実験 2：対応つけていない二言語の既存文書の同期化：英中 既存の 5 つの多言語文書（英語版と中国語版がある）M4-M6, D3-D4 を利用して，英語版と中国語版の同期化実験を行った．英語と中国語が出来る 10 人を SynCLDoc を利用するチーム（5 人）と，System B を利用するチーム（5 人）に分けて，実験した．M4 と D3 の各言語の版は完全一致で（各言語の版が 50 文で 10 段落からなる），M5 と D4 の各言語の版にそれぞれ違うところがあり（各言語の

[*3] 日本語能力試験認定の目安を参考してください．http://www.jlpt.jp/about/levelsummary.html
[*4] 中国の英語能力テストである．TOEIC 試験の 750 分ぐらいのレベルである．

表 5.6. 実験 1 ：多言語文書の英語版と日本語版を同期する結果

文書	SynCLDoc の同期化時間	見逃し率（SynCLDoc）	System B の同期化時間	見逃し率（System B）
M1	18 分	0%	21 分	0%
M2	23 分	0%	19 分	2%
M3	21 分	0%	25 分	0%
D1	35 分	0%	28 分	8%
D2	38 分	0%	32 分	12%

表 5.7. 実験 2 ：多言語文書の英語版と中国語版を同期する結果

文書	SynCLDoc の同期化時間	見逃し率（SynCLDoc）	System B の同期化時間	見逃し率（System B）
M4	15 分	0%	12 分	0%
M5	26 分	0%	20 分	6%
M6	23 分	0%	21 分	0%
D3	32 分	0%	26 分	4%
D4	40 分	0%	33 分	6%

版が 50 文で 11 段落からなる．全部 10 ヶ所が違う），M6 の各言語の版の内容はまったく一致しない（各言語の版が 50 文からなる）．また，実験結果も実験 1 と同じで（5.7），SynCLDoc を利用する時と利用しない時，対応つけていない多言語文書の二言語の版の同期化する時間は大きな差が存在しなかった．

■実験 3：対応つけていない三言語の既存文書の同期化　既存の 5 つの多言語文書（日本語版，英語版と中国語版がある）M7-M9, D5-d6 を利用して，同期化実験を行った．日本語，英語と中国語が出来る 10 人を SynCLDoc を利用するチーム A（5 人）と，System B を利用するチーム B（5 人）に分けて，実験した．多言語文書 M7-M9, D5-D6 の同期化作業は一度もされていなかった．つまり，対応関係が完全に存在しない．M7 と D5 の各言語の版は 10 段落で，合計 50 文があり，また，内容は完全一致である．M8 と D6 の各言語の版はそれぞれ違うことろがある．英語版と日本語版の違うところが 10 ヶ所があり，英語版と中国語版の違うところが 10 ヶ所があり，中国語版と日本語版の違うところが 5 ヶ所がある．つまり，日本語版と英語版の差分の一部と英語版と中国語版の差分の一部は日本語版と中国語版に存在しない．例えば，日本語版の一段落目に「テストです.」という文があり，中国語版の一段落目に「是一個実験.」という文があるが，英語版の一段落目に「It is a test.」という文がない場合，ユー

ザーは英語版の一段落目に「It is a test.」という文を追加したら，三つの言語の版が同期できる．従って，3 言語以上の言語の版がある多言語文書の同期化は，より複雑である．M9 の日本語版と中国語版がまったく同じで，日本語版，中国語版と英語版は完全に一致しない（各言語の版が 50 文からなる）．

表 5.8 に示すように，SynCLDoc を利用した三言語以上の言語の版がある多言語文書の同期化は System B を利用した同期化作業より，速かった．また，差分の判断の正確性も高かった．この原因は SynCLDoc は既存言語ペア間の対応関係を利用し，新たな言語ペア間の対応関係の判定ができるためである．実験 3 を行うとき，まずチーム A の人々が SynCLDoc を利用し，多言語文書の英語版と日本語版を対応付け，同期化した．チーム B の人々が System B を利用し，多言語文書の英語版と日本語版を対応付けて，同期化した．次に，チーム A の人が SynCLDoc を利用し，多言語文書の英語版と中国語版を対応付け，同期化した．チーム B の人々が System B を利用し，多言語文書の英語版と中国語版を対応付け，同期化した．そのとき，チーム A の人々がかかった時間とチーム B の人々がかかった時間が同じぐらいであった（実験 1 と実験 2 の結果と同じ）．そして，両チームは中国語版と日本語版の同期化を行う．そのとき，多言語文書 M7 と D5，多言語文書 M8 と D6 と，多言語文書 M9 それぞれの作業が異なる．

■M7 と D6 と M9 の場合 M7 と D6 の各言語の版が完全一致であり，また M9 の日本語版と中国語版がまったく同じなので，中国語版と日本語版の同期化する作業が要らない．実験 3 を行うとき，三つの言語の版の同期化は同じ人で行うため，SynCLDoc を利用しても，System B を利用しても，ユーザーが英語版と日本語版，英語版と中国語版が完全一致ということが分かり，日本語版と中国語版の同期化が必要がないことが分かった．また，M9 の日本語版と英語版を同期作業を行い，中国語版と英語版の同期化作業をする時，新しい修正がないため，日本語版と中国語版の同期化が必要がないことが分かった．

今回一人のユーザーが一つの多言語文書の三つの言語の版の同期実験を行ったが，二人以上のユーザーが協力的にある多言語文書の三つの言語の版の同期を行うと，違う結果が出る可能性があると考えられる．複数人でそれぞれの言語ペア間の同期化作業を行う場合，System B を利用した場合，日本語版と中国語版を同期化する人は他の言語ペア間（例えば，英日や英中）の同期化を行わない場合も可能なので，日中の内容が完全一致することを分からず，日本語版と中国語版をもう一度同期化作業を行わなければならない．そのため，このような場合は，ユーザーは日本語版と中国語版の同期化も初めに二つの言語の版の同期化作業と同じくらいの時間がかかると想定できる．つまり，System B を利用する場合，M7 や M9 や D6 のような他言語文書を同期化する場合，実験結果により多くの時間がかかる可能性が存在する．

一方，SynCLDoc は一回付けた言語ペア間の対応関係を保存し，新たな言語ペア間の対応関係を判定し，ユーザーに示すことができる．複数人でそれぞれの言語ペア間の同期化作業を行っても，あるユーザーは他のユーザーたちの同期化作業の確認ができる．そのため，上記のような場合にも，他のユーザーがすでに英日と英中の対応関係を付け，確認したことが分かり，もう一度日中の対応関係を付けることがしなくても良いことがわかる（節 5.2 参照）．

表 5.8. 実験 3 ：対応つけていない三言語の既存文書の同期化結果

文書	SynCLDoc の同期化時間	見逃し率（SynCLDoc）	System B の同期化時間	見逃し率（System B）
M7	41 分	0%	38 分	0%
M8	48 分	0%	79 分	10%
M9	53 分	0%	51 分	2%
D5	45 分	0%	43 分	4%
D6	57 分	0%	92 分	16%

従って，SynCLDoc を利用する場合，違う人で多言語文書の同期化をしても，多くの時間がかからない．

■M8 と D6 の場合　M8 と D6 の各言語の版の内容はそれぞれが違うところがあるので，中国語版と日本語版の同期化する作業が要る．ユーザーは英語版と日本語版を同期してから，英語版と中国版を同期したら，英語版と中国が最新になるが，日本語版がまた古くなった．なので，中国語版と日本語版（または英語版と日本語版）の同期化作業が必要である．SynCLDoc を利用する場合，システムは日本語版と英語版，英語版と中国版の対応関係を利用し，日本語版と中国語版の対応関係を判定し，ユーザーに示す．ユーザーは SynCLDoc が計算した対応関係は既存の言語ペア間から取得したことが分かり（SynCLDoc はこのメッセージを表示），対応関係の確認と修正が楽にできた（節 5.2 参照）．また，差分の確認と差分の対応箇所もすぐに分かり，同期しやすい．

■実験 4：対応つけた多言語文書を修正し，同期化する　既存の 16 つの多言語文書 M1-M3,M7-M12,D1-D2,D5-D9 を利用して，多言語文書に修正を与え，日本語版，中国版と英語版の同期化実験を行った．

多言語文書 M1-M3,M7-M12,D1-D2,D5-D9 の英日，英中，日中の正確な対応関係が全部存在する．日本語，英語と中国語が出来るユーザー 10 人は全部 M1-M3,M7-M9,D1-D2,D5-D6 の多言語文書の同期化を一回やったことがあったが，M10-M12, D7-D9 は読んだことがなかった．本研究は多言語文書 M1-M3,M7-M12,D1-D2,D5-D9 の任意の版にいくつかの修正をして，ユーザー 10 人に同期化作業を与えた．チーム A の 5 人は SynCLDoc を利用し，チーム B の 5 人は System B を利用した．

表 5.9 に示すように，同期化された多言語文書を修正を与え，再同期化する作業を行う場合，SynCLDoc を利用した同期化作業がかかった時間は System B を利用した同期化作業がかかった時間より，大幅に短かった．なぜなら，SynCLDoc は多言語文書の対応関係と類似度計算により，差分の特定と差分が他の言語の版の対応箇所が特定でき，ユーザーに分かりやすく表示することができるためである．ユーザーは差分の対応箇所が正しいかどうかを判断し，

表 5.9. 実験 4：多言語文書を修正し，同期する結果（対応関係が存在）

修正（修正総数）	同期化平均時間（チーム A）	同期化平均時間（チーム B）
日本語版のみ修正（5）	2 分	11 分
日本語版，中国語版違う修正（5）	3 分	16 分
日本語版，中国語版同じ修正（5）	1 分	7 分
三国語版同じ修正（10）	0 分（必要がない）	13 分
三国語版違う修正（10）	4 分	21 分

翻訳作業を行えば，同期化できる．SynCLDoc が特定された対応箇所はほぼ正しく（少なくとも，段落の対応が正確），ユーザーが差分の対応箇所を特定する時間が大幅に削減できた．また，違う言語の版にそれぞれ修正を与えた場合，SynCLDoc はこれらの修正は同じ修正かどうかを判断し，ユーザーにどの言語の版の同期化作業が必要かを示す．ただし，SynCLDoc により，同じ修正なのに，判断できない場合，または違う修正が同じ修正に判断した場合も存在したので，履歴の確認で差分の判断が正しいかどうかを確認する必要がある場合も存在する．一方，System B を利用した場合，ユーザーは修正内容が分かっても，その修正内容が他の言語の版のどこに対応するかが分からなく，対応箇所を特定する時間が長くかかった．また，違う言語の版にそれぞれ修正を与えた場合，System B はこれらの修正は同じ修正かどうか判断できないため，各言語の版の状態（つまり，更新が必要かどうか）はユーザーが手作業で判断する必要があるため，余計に時間がかかった．

実験 4 に利用した多言語文書は 50 文くらいの文書のため，System B を利用した同期化作業の時間はそれほど多くないが，もし，多言語文書の各言語の版が長い場合，System B を利用する場合，ユーザーは手作業で差分の判断や修正箇所の特定などを行う時，もっと時間がかかる．一方，SynCLDoc を利用する場合は，ユーザーはシステムが判定された差分の対応箇所が正しいかどうかを判断するだけで，同期化作業の時間が修正の数だけの影響を受け，大幅に長くなる状況が発生しないと考えられる．

5.4 本システムの有用性について

本研究の目標は，複数人で多言語文書の同期化を効率的に実現することである．この目標を達成するために，異なる言語文書間の差分の対応付けにより，修正箇所を特定し，ユーザーに分かりやすく表示するシステム SynCLDoc を開発した。SynCLDoc は多言語文書間の段落と文の対応関係を計算することにより，ある言語の文章を修正した時，他の言語の文書の対応箇所を特定し，強調して表示することができる．

既存の類似度計算アルゴリズムを利用した二つの言語の版の対応関係の計算の評価により（節 5.1），本研究が開発した SynCLDoc はマニュアルのような分かりやすく書かれた文章に

対応関係の計算結果が良く，ユーザーによる対応関係の修正の手間が少ないが，小説のような文学性が強い文書やニュースなどの新語や固有名詞が多い文書に対してはシステムによる自動的に対応関係の計算結果の正確率が低く，ユーザーによる対応関係の修正の手間がかかる．

また，他の言語ペアを活用した新たな対応関係の判定の評価により（節 5.2），SynCLDoc は速く，正確に新しい言語ペア間の対応関係が取得し，ユーザーによる対応関係の修正の手間を減らすことができる．例えば，ユーザーが多言語文書の日本語版と英語版の対応関係と，英語版と中国語版の対応関係を修正したら，その結果を日本語版と中国語版の対応関係にフィードバックするようにした．従って，ユーザーは日本語版と中国版の対応関係を確認するとき，比較的に楽になる．これは特に，類似度計算による対応関係が正確に付けない小説などの 3 言語以上の版を持つ多言語文書は，対応関係確認にかかる時間を大幅に削減できる．

システム実験による SynCLDoc のユーザー評価により（節 5.3），SynCLDoc を利用すると，多言語文書の同期化をより高速にできる．初めに多言語文書の二つの言語の版を同期するときは，差分を取るため，対応関係の確認と修正が必要で，システムを利用しなく差分を取る方法より，少し多くの時間がかかる場合もある．特に，各言語の版の内容がまったく一致しない場合や，小説などの類似度計算により取得した対応関係の正確率が低い場合，初めに同期化をする時，SynCLDoc を利用しても，差分を取る時間をあまり減らせない．しかし，マニュアルのような分かりやすく書かれた文章や，3 言語以上の版を持つ多言語文書を同期すると，SynCLDoc を利用して差分を取るのにかかる時間はシステムを利用しなくて差分を取るのにかかる時間より短くなる．また，一度ユーザーが多言語文書の各言語の版の対応関係を修正し，確認すると，修正が行われた時，SynCLDoc によって，多言語文書の各言語の版の状態をすぐ確認でき，修正内容とその修正が他の言語の版のどこに対応するかもすぐに分かり，探す手間がほぼなくなる．多言語文書の言語の種類が多いほど，各言語の版の状態と修正内容の対応する箇所の特定がより多くの時間と手間がかかるため，SynCLDoc は 3 言語以上の版がある多言語文書に頻繁な修正を与える場合，特に有効である．

ただし，今回システム実験を行ったユーザーは全部中国人で，そして，二つの言語や三つの言語ができる人である．母語が他の言語の人や，一つの言語しかできない人や，同じ中国人で，日中英三つの言語ができるけど，レベルが違う人などのユーザーがシステム実験を行ったら，違う結果が出る可能性もある．例えば，実験 3 のテスト環境で，二つの言語しかできる人と三つも言語ができる人の実験結果が明らかに違うと考えられる．例を挙げると，M7 のような各言語の版の内容が完全一致で，複数の二つの言語しかできる人たちと一人の三つも言語ができる人が同期化作業を行うと，従来の同期化支援システムを利用する場合，複数の二つの言語しかできる人たちは一人の三つも言語ができる人より三分の一くらい多くの時間がかかることが推測できる．その一方，SynCLDoc を利用したら，複数の二つの言語しかできる人たちと一人の三つも言語ができる人は同じぐらいの時間がかかることが推測できる．

既存の多言語文書を SynCLDoc を利用し，同期化する作業を行う時，初めに言語の版をシステムに投入したら，ユーザーが対応関係の確認が必要で，初期時間がかかるが，SynCLDoc を利用し，多言語文書を作成する場合，各言語の内容が段々追加するため，ユーザーの初期確認作業が要らなくなり，より同期しやすくなると考えたらる．また，SynCLDoc はソフト

ウェアマニュアルのような頻繁に修正を与える多言語文書，または言語種類が多数存在する多言語文書の同期化を特に効率的に支援できる．

SynCLDoc は現在各言語の版の対応関係を計算するには少し時間がかかる（100 文がある二つの文書の対応関係の計算が約 1 分間くらいかかる）が，それは実装の問題で発生した．対訳データベースの構造をよくすれば，解決できる．また，日本語，英語，中国語しか支援できないが，支援する言語を増加することは容易である．そして，提案された文の類似度計算による対応関係の計算アルゴリズムは単純ではあるが，マニュアルのような多言語文書の同期化の支援には効果的である．この対応関係の計算アルゴリズムには，まだ改良の余地があり，より効果的に多言語文書の同期化を支援できると考えられる．

第 6 章

結論

本研究の目的は，多言語文書の異なる言語の版の差分の対応付けにより，差分判断と対応箇所を特定し，ユーザーに分かりやすく表示することで，多人数で多言語文書の同期化を効率的に実現することである．複数人で多言語文書をメンテナンスする需要が日々高まっているため，多言語文書の同期化という新しい問題が生じた．それは，既存の多言語文書同期化支援ツールでは，ある言語の文章を修正した時，他の言語の文書の修正箇所の把握が困難であった。特に，各言語の版に独立した編集内容を頻繁に加える文書に対して，各言語の版の修正内容とその修正内容の他の言語の版の対応箇所の特定がより困難であった．そこで，本研究は，文の類似度計算アルゴリズムを利用し，多言語文書の各言語の版の段落と文の対応関係を計算することにより，異なる言語の版の差分と対応箇所を特定し，強調して表示するシステム SynCLDoc を開発した。SynCLDoc は各言語の版の内容に独立した編集を頻繁に加える多言語文書を対象として開発された．例えば，オープンソースのソフトウェアのマニュアル．Wikipedia のような各言語ページの内容は同じ内容を書く必要がない文章や，小説や記事などの修正頻度が少ない多言語文書は対象外である．

既存の多言語文書の同期化を支援するツールでは，効率的に多言語文書の言語の版の差分を取り，その差分が他の言語の版のどこに対応するかが特定できない．ある言語の版の内容を修正するとき，他の言語の版の対応する位置を自動的にユーザーに示すことで，この複雑な差分の対応箇所の特定作業が要らなくなると考えられる．これを実現するため，本研究は既存の文の類似度アルゴリズムを基づいて，多言語文書の異なる言語の版の対応関係を計算し，多言語文書の差分とその差分の対応箇所を特定するシステム SynCLDoc を設計した．複数人が同時に利用できるため，ウェブアプリケーションとして実装を行った．

まずは既存の文の類似度計算アルゴリズムを利用し，段落と文の対応関係を計算する方法を提案した．文の類似度計算アルゴリズムによる計算された多言語文書の対応関係だけでは，うまく対応付けができなかったため，段落と文の順番なども考えて，対応関係を計算する方法実装した．評価により，このアルゴリズムによる計算したソフトウェアのマニュアルのような多言語文書の段落の正確な対応率が 95% 以上であり，また，文の正確な対応率が 91% 以上であることを確認した．しかし，既存の文の類似度アルゴリズムは小説などの文学性が強い文にはよく効かないこともわかった．

また，言語 A,B に対するすでに計算された対応関係 R(A,B) から，新たな対応関係を計算する方法を二つ提案した．一つ目は修正内容の対応箇所を古い対応関係から計算する方法を実装し，修正内容の対応箇所の特定ができた．二つ目は，複数の言語の版 L1, L2, L3 について，すでに計算された対応関係 R(L1,L2)，R(L1,L3) から，新しい対応関係 R(L2,L3) を計算する方法を実装し，より速く，正確に対応関係の取得ができた．評価により，すでに計算した対応関係 R(L1,L2)，R(L1,L3) から，新しい対応関係 R(L2,L3) を計算する場合，効率的に正確な対応関係 R(L2,L3) を取得することが確認できた．ただし，L1，L2，L3 の文がどれくらい R(L1,L2)，R(L1,L3) に存在するかの影響を受ける．例えば，L1 の文と L2 の文が全部 R(L1,L2) に存在し，L2 の文と L3 の文が全部 R(L2,L3) に存在するとき，R(L2,L3) が 98% 以上正しく取り出すことが確認したが，L1 の文と L2 の文が 80% くらい R(L1,L2) に存在し，L2 の文と L3 の文が 80% くらい R(L2,L3) に存在するとき，R(L2,L3) が 95% 以上正しく取り出すことが確認できた．

さらに，計算した対応関係を用いて，同期が必要な箇所をわかりやすく表示する手法を実装した．本研究が行ったシステムユーザー実験によると，SynCLDoc が計算した対応関係を確認する必要があるが，一度その確認作業が行われれば，その作業の結果をもとに，より速く，正確に修正内容と対応箇所の特定を行うことが確認できた．特に，ソフトウェアマニュアルのような頻繁に修正が加わる多言語文書の同期化作業における差分の特定の手間は大幅に削減できることが確認できた．初めに多言語文書を同期化するとき，SynCLDoc を利用する場合と従来の同期化支援システムを利用する場合，同期化作業にかかる時間が同じくらいであった．しかし，三つの言語がある多言語文書を同期化する場合，SynCLDoc を利用する場合，かかる時間は従来のシステムを利用する場合より，約 1/3 の削減ができた．また，同期化された多言語文書を修正し，再同期化する場合，SynCLDoc を利用したら，約 80% の差分の特定時間が削減できた．

最後に今後の課題について述べる．類似度計算アルゴリズムは対訳データベースの依存が強く，また，小説やニュースのような意訳の内容が多い多言語文書の対応関係の計算が弱いため，文の類似度による対応関係の計算だけでなく，文間の距離などの方法を利用し，対応関係の計算するアルゴリズムを改善する方法が今後の課題である．また，翻訳メモリなどの機能をシステムに導入し，より高速に，正確に対応関係の計算や各言語の版の修正内容が同じかの判定を行うことも今後の課題である．そして，SynCLDoc は現在，多言語文書をシステムにアップロードし，同期化する作業と編集作業もすべて SynCLDoc 上で行わないと，できないが，今後に SynCLDoc を利用し，同期化作業を行えば，ローカルに同期化された多言語文書にダウロードし，編集作業が行えることも一つの課題である．さらに，現在のシステムはテキストのみしか対応しておらず，マニュアルのような多言語文書には画像なども存在するので，画像なども扱えるようにすることも今後の課題である．

発表文献と研究活動

(1) Wenting Gu, Koichi Sasada, Shigeru Chiba. A Repository System for Cross-lingual Documents, 29th JSSST Conference, Augest, 2012

参考文献

[1] Brent Hecht and Darren Gergle. The tower of babel meets web 2.0: user-generated content and its applications in a multilingual context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 291–300, New York, NY, USA, 2010. ACM.

[2] Louis-Philippe Huberdeau, Sebastien Paquet, and Alain Desilets. The cross-lingual wiki engine: enabling collaboration across language barriers. In Ademar Aguiar and Mark Bernstein, editors, *Int. Sym. Wikis*. ACM, 2008.

[3] Ching man Au Yeung, Kevin Duh, and Masaaki Nagata. Providing cross-lingual editing assistance to wikipedia editors. In *CICLing (2)*, pp. 377–389, 2011.

[4] Christof Monz, Vivi Nastase, Matteo Negri, Angela Fahrni, Yashar Mehdad, and Michael Strube. Cosyne: a framework for multilingual content synchronization of wikis. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, WikiSym '11, pp. 217–218, New York, NY, USA, 2011. ACM.

[5] 将夫内山, 均井佐原. 日英新聞の記事および文を対応付けるための高信頼性尺度. 自然言語処理 = Journal of natural language processing, Vol. 10, No. 4, pp. 201–220, jul 2003.

[6] Karolina Owczarzak, Josef Genabith, and Andy Way. Evaluating machine translation with lfg dependencies. *Machine Translation*, Vol. 21, No. 2, pp. 95–119, June 2007.

[7] Hua Wu and Haifeng Wang. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, Vol. 21, No. 3, pp. 165–181, September 2007.

[8] Simon Carter and Christof Monz. Syntactic discriminative language model rerankers for statistical machine translation. *Machine Translation*, Vol. 25, No. 4, pp. 317–339, December 2011.

[9] Vassilina Nikoulina, Bogomil Kovachev, Nikolaos Lagos, and Christof Monz. Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pp. 109–119, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[10] Yanjun Ma, Yifan He, Andy Way, and Josef van Genabith. Consistent translation using discriminative learning: a translation memory-inspired approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pp. 1239–1248, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[11] Sharon O'Brien, Minako O'Hagan, and Marian Flanagan. Keeping an eye on the ui design of translation memory: how do translators use the "concordance" feature? In *Proceedings of the 28th Annual European Conference on Cognitive Ergonomics*, ECCE '10, pp. 187–190, New York, NY, USA, 2010. ACM.

[12] Sarah Dillon and Janet Fraser. Translators and tm: An investigation of translators' perceptions of translation memory adoption. *Machine Translation*, Vol. 20, No. 2, pp. 67–79, June 2006.

[13] Ignacio Garcia. Power shifts in web-based translation memory. *Machine Translation*, Vol. 21, No. 1, pp. 55–68, March 2007.

[14] Chang Hu, Benjamin B. Bederson, Philip Resnik, and Yakov Kronrod. Monotrans2: a new human computation system to support monolingual translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 1133–1136, New York, NY, USA, 2011. ACM.

[15] Yamabana Kiyoshi, Muraki Kazunori, Kamei Shin-ichiro, Satoh Kenji, Doi Shinichi, and Tamura Shinko. An interactive translation support facility for non-professional users. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pp. 324–331, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[16] Nic Bertino. Modern version control: creating an efficient development ecosystem. In *Proceedings of the ACM SIGUCCS 40th annual conference on Special interest group on university and college computing services*, SIGUCCS '12, pp. 219–222, New York, NY, USA, 2012. ACM.

[17] Maurício Massaru Arimoto, Maria Istela Cagnin, and Valter Vieira de Camargo. Version control in crosscutting framework-based development. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pp. 753–758, New York, NY, USA, 2008. ACM.

[18] Jennifer Vesperman. *Essential CVS*. O'Reilly Media, Inc., 2003.

[19] Christopher C. Yang and Kar Wing Li. Building parallel corpora by automatic title alignment using length-based and text-based approaches. *Inf. Process. Manage.*, Vol. 40, No. 6, pp. 939–955, November 2004.

[20] Jason R. Smith, Chris Quirk, and Kristina Toutanova. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Associ-*

ation for Computational Linguistics, HLT '10, pp. 403–411, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[21] Xiaoyi Ma. Champollion: A robust parallel text sentence aligner. In *In Proceedings of LREC-2006*, 2006.

[22] Min-Hsiang Li, Vitaly Klyuev, and Shih-Hung Wu. Multilingual sentence alignment from wikipedia as multilingual comparable corpora. In *Proceedings of the 13th International Conference on Humans and Computers*, HC '10, pp. 167–171, Fukushima-ken, Japan, Japan, 2010. University of Aizu Press.

[23] Takehito Utsuro, Hiroshi Ikeda, Masaya Yamane, Yuji Matsumoto, and Makoto Nagao. Bilingual text matching using bilingual dictionary and statistics, 1994.

[24] Sentences Masao Utiyama, Masao Utiyama, and Hitoshi Isahara. Reliable measures for aligning japanese-english news articles and. In *in Proceedings of the 41st Annual Meeting of the ACL*, pp. 72–79, 2003.

[25] Takehito Utsuro, Takao Shime, Masatoshi Tsuchiya, Suguru Matsuyoshi, and Satoshi Sato. Learning dependency relations of japanese compound functional expressions. In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*, MWE '07, pp. 65–72, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[26] Yotaro Watanabe, Junta Mizuno, Eric Nichols, Katsuma Narisawa, Keita Nabeshima, Naoaki Okazaki, and Kentaro Inui. Leveraging diverse lexical resources for textual entailment recognition. Vol. 11, No. 4, pp. 18:1–18:22, December 2012.

[27] Toru Hisamitsu and Yoshihiko Nitta. Analysis of japanese compound nouns by direct text scanning. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pp. 550–555, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[28] Jingyang Li, Maosong Sun, and Xian Zhang. A comparison and semi-quantitative analysis of words and character-bigrams as features in chinese text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pp. 545–552, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[29] Christopher C. Yang and Kar Wing Li. Error anaylsis of chinese text segmentation using statistical approach. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, pp. 256–257, New York, NY, USA, 2004. ACM.

[30] Richard Sproat, William Gale, Chilin Shih, and Nancy Chang. A stochastic finite-state word-segmentation algorithm for chinese. *Comput. Linguist.*, Vol. 22, No. 3, pp. 377–404, September 1996.

謝辞

本研究は多くの方々のご指導，ご支援のおかげで論文としてまとめることができました．

指導教員である東京大学の千葉滋教授と元指導教員である HeroKu, Inc. の笹田耕一氏には，日頃から親切に熱心にご指導いただきました．特に笹田耕一氏には，夜遅くまで秋葉原にて議論にお付き合いいただき，研究の方向性を決める上で大変参考になるアドバイスをいただきました．厚くお礼を申し上げます．また大学内外の方々と議論できる場を積極的に設けていただき，学問に限らず様々なことを学ばせていただきました．深く感謝いたします．

また，中国にいる楼元傑さん，蒋茜茜さん，呉悦倩さん，沈靖さん，王思宏さん，朱江涵さん，曹燕華さん，朱萍さん，唐旭嵐さん，姜虹さん，陳　さん，張燕さん，陳校校さん，そして元富士国際語学院の孫ナイキさん，陳静秋さん，周セイキさん，王青さん，趙一星さんは本研究のシステムの実験を行い，多くの時間をいただきました．深く感謝いたします．また，トランスコスモス株式会社の呉晟毅さんは時間を作り，本稿の用語などを修正していただきました．深く感謝いたします．

最後に，元笹田研究室の皆様と現在千葉研究室の皆様には，研究に対して多くの助言をいただいたり，議論をしていただきました．研究以外でもシステム開発に関すること，プログラミングに関することなど，いろいろなことをとても楽しく勉強させていただきました．外国人の私に色々教えて，助けてくれて，修士二年間を楽しく過ごせた．深く感謝いたします．