公告本

304253

申請日期	85.6.19
案號	85107412
類別	G06F 9/45, 17/00

Int.Cl[6]

A4
C4 304253

（以上各欄由本局填註）

發明 新型 專利說明書

一、發明 新型 名稱	中文	機器輔助的翻譯工具
	英文	Machine Assisted Translation Tools
二、發明 創作 人	姓名	1.詹姆士E.哈格雷夫三世 (James E. Hargrave, Ⅲ) 2.伊貝斯I.薩白瑞爾 (Yves I. Savourel)
	國籍	1.美國 2.美國
	住、居所	1.美國科羅拉多州80501龍格蒙特第十八街1821號 2.美國科羅拉多州80302布爾德#253佳揚2227號
三、申請人	姓名（名稱）	國際語言工程股份有限公司 International Language Engineering Corporation
	國籍	美國
	住、居所（事務所）	美國科羅拉多州80301布爾德藍吉街1600號
	代表人姓名	伯納德加特歐(Bernard Gateau)

裝 訂 線

經濟部中央標準局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

（由本局填寫）

承辦人代碼：	A6 B6
大　　類：	
ＩＰＣ分類：	

本案已向：

美　國（地區）　申請專利，申請日期：1995年 6月 7日　案號：08/484,981　，□有　□無主張優先權

有關微生物已寄存於：　　，寄存日期：　　，寄存號碼：

（請先閱讀背面之注意事項再填寫本頁各欄）

裝　訂　線

五、發明說明（1）

發明背景

1.發明領域

本發明有關文字及語言之機器處理，且較特別地，有關一種含有軟體來實現「機器輔助翻譯」的方法及設備。

問題說明

從一個語言翻譯其文字至另一語言，通常是一個需要有經驗的翻譯人員之單調且乏味的工作。自從電腦的時代來到之後，專家們開始利用電腦為輔助工具，以進行「自然語言」(natural language)的翻譯。最早的機器翻譯MT(machine translation)系統，依賴厚厚的雙語字典中，根據原始語言SL(SOURCE LANGUAGE)，所找出目標語言TL(target language)的一個或多個相對應的解釋。很快地，專家們了解到字典裡面，關於語法及文法的規則過於複雜，實在沒有辦法發展出一套描述（任何）語言的詳盡規則。事實證明這些問題很難處理，以至於專家們對機器翻譯MT所作過的努力，大都已經被放棄了。

放眼全球，多語言的多元文化以及國際貿易，使得翻譯服務的需求，日益增加。在翻譯服務的市場，商業文件及技術文件的翻譯需求，佔有一相當並日益增加的分量。合約、指導手冊、表格以及電腦軟體之類的文件，均是很好的例子。當一個產品或是一項服務被"本地化"(localized)，以被推廣到一個新的國外市場時，通常會有許多文件需要翻譯，因而突出了經濟的翻譯服務之

五、發明說明（ ）

重要性。因為商業及技術資訊通常都是很詳細而精確的，準確的翻譯也會繼續有其需求。

機器翻譯MT系統通常可以分為：「直接式」(direct)、」轉移式」(transferbased)或是「介於多種語言式」(interlingua-based)。於「直接式」的機器翻譯中，在原始語言和目標語言之間，沒有「中介表示方式」(intermediate representations)。原始語言的文字"直接"被處理，以轉換成目標(語言)的文字。除了少許調整，基本上此方式逐字翻譯。因為這種方式，先天俱有忽略各種句子內部結構形態的缺陷，今天的機器翻譯MT系統，已經不再使用此方式了。

「轉移式」的翻譯方式會分析原始語言的文字，並將各步驟分析出的資訊，轉移給產生目標語言文字的相對應之步驟。例如，它可在字彙、文法或是由文法所建立的(句子)結構等階段，做相對應的轉移（由原始語言到目標語言）。因為這種轉移方式，僅適用於一對特定的語言（一個特定的原始語言和一個特定的目標語言），每一對不同而需要翻譯的語言，都必須特別而辛苦的建立有轉移方式。

「介於多種語言式」的方法，假設一個合適的「中介表示方式」可以被定義出，所以原始文字可先被對應到該「中介表示方式」，再由該表示方式對應到目標文字。基本上，此一方式比較有吸引力，它不需要像「轉移式」的方式，為每一對原始及目標語言建立一個個別的

五、發明說明（ 7 ）

轉移程式。然而一個真正與語言無關（language independent）的「中介表示方式」，是否可以發展出，仍是一個疑問。今天現有的「介於多種語言式」翻譯系統，對於其「中介表示方式」的通用性（通用於各種語言）之宣稱，仍然是非常的薄弱。為達到高品質的翻譯水準，原始和目標語言之間，某些特定的形態仍然時常需要被考慮到。

「轉移式」的翻譯方式，最近有些進步。在發展文法的數學及計算模型之過程中，將字彙的結構歸類，並定義出組成這些物件的運算，以從字彙項目中直接找出語法（syntactic）和語意（semantic）資訊的重要性，越來越受到強調。從這個角度來看，所有與某一語言特定有關的資訊，都會被包括在此字彙項目，以及其所附的結構內。在這種方式下，不同的語言在此階層會有所區別，但是對於組成這些結構的運算，所有的語言都是相同的。這種翻譯方式下一步的目標，是在本階層上，定義出所有雙語（原始和目標語言）之間的關連。本翻譯方式是否能夠在許多語言上使用，仍然是一個值得探討的課題。

有些現有的機器翻譯MT系統，對於被翻譯的原始文件之撰寫方式上，有高度的規定要求。這種系統對於準備不同語言的手冊的確有幫助。但是這種系統並不是在翻譯一個以「自然語言」所撰寫的手冊，到許多其它的「自然語言」。它們僅能算是從一個依據高度規格限制所撰寫出的手冊，產生出許多其它語言的文字，因而避免了許多

五、發明說明（4）

傳統機器翻譯MT上的問題。

近來，專家們注意到利用機器來輔助翻譯人員的各種方式，而不是讓機器自主的負擔翻譯工作。這種方法稱為「機器輔助翻譯」(machine assisted translation)或是「交談式翻譯)(interactive translation)。現在已有可翻譯出高品質商業書信的系統。其原理是利用由機器翻譯出的段落，配合翻譯人員的修飾。「翻譯記憶庫」TM(translation memory)是一個機器輔助翻譯工具的例子。翻譯記憶庫是一個資料庫，在翻譯時它會收集翻譯出的文件及其原始語言的版本。在翻譯過數份文件後(同時亦將這些文件收集到資料庫之後)，此工具可以用來協助新的翻譯工作。如果原始語言含有與翻譯記憶庫內相同或是相似的文字，先前翻譯出的文字，可以協助新的翻譯工作。

從理論上來說，這種系統的優點是它可以利用現有的機器翻譯MT技術，提高翻譯人員的效率，而又不喪失傳統以來翻譯人員所提供的準確翻譯品質。因為它確保翻譯人員永遠不需要重複翻譯相同的原始文字，它可提高翻譯的效率。然而翻譯記憶庫需要很大的資料檔案，且需要搜尋以找回相符合的文字，其執行速度很慢。通常有經驗的翻譯人員翻譯之速度，要比翻譯記憶庫找出以前翻譯過的文字之速度還要快。對於翻譯記憶庫工具而言，快速的文字搜尋以及找回能力，尚有待加強。

翻譯記憶庫最大的用途不僅是找出以往已經翻譯過完

五、發明說明（5）

全相同的文字，亦包括「近似」（approximate）或是「模糊匹配」（fuzzy matches）的文字。「模糊匹配」的功能使得稍有不同的文字亦能被找出，包括順序、形態、大小寫以及拼法稍有不同的文字。因為「自然語言」文字中所具有的無限可能（不同的組合），「近似」的功能也是需要的。利用「模糊匹配」的系統包括Trados所出版的Translator's Workbench for Windows以及Atril所出版的Deja Vu。值得一提的是這種系統的性能，與其「模糊匹配」實行的方式有相當重要的關連。

因為翻譯記憶庫TMs不會去分析語言及文法，和其它的翻譯技術比較起來，它和（所翻譯的）語言之間較為獨立，沒有太大的關連。但是在實際的應用上，發展出真正和語言獨立無關的搜尋軟體，仍然是很困難的。更明確的來說，現有搜尋技術的核心，仍是建立在「字」的基礎上，所以它們工作的原理，必需依賴「字」作為其基本的元素。此一情形對於「模糊」搜尋（fuzzy search）的方式，尤其顯著。在每個語言中，文字都會依性別、單複數以及時態的情形，有其獨特的變化。因為文字與其所屬的語言，有不可分割的關連，以「字」為基礎的翻譯系統，無法真正的獨立於（所翻譯的）語言之外。發展出一個快速而準確的「模糊」文字搜尋方式，一直是很困難的。

「索引」（concordances）是翻譯人員常用的另一種工具。電子索引代表的是包含有「字串」以及其出現在文

五、發明說明（ b ）

件中，上下文意思之檔案。這裡所指的字串可能包含文字、片語以及句子。當翻譯人員不確定要為某一字採用何種意義時，索引可以列出該字在不同上下文中的各種不同意義。這種資訊可以協助翻譯人員，更準確的選出恰當的翻譯用詞，以符合其在原始語言文件中的意義。電子索引具有搜索文字的軟體，能夠讓翻譯人員根據一個所要的字或是片語，從其資料庫中找出所有包含這個字或是片語的文字字串。快速的瀏覽一下之後，所找出的文字字串可以幫助翻譯人員更加瞭解某一字或是片語在該上下文的用法。因為速度太慢，或是「模糊」搜尋（fuzzy searching）能力的不足，直到今天為止，翻譯記憶庫仍然不能支援索引的搜尋。

在國際商業活動以及通訊的領域，多語種「自然語言」的處理，有其日益增加的重要性和發展的機會。為提高文件翻譯的效率，並降低其成本，機器輔助之翻譯工具是需要的。為了能有效利用過去已翻譯出的商業及技術文件，所累積儲存下來的大量知識，亦需要機器輔助的翻譯工具。更明確的來說，翻譯的領域需要一個和語言獨立無關的翻譯記憶庫工具，以能從過去已經翻譯出的文件中，快速而準確的提供「模糊」（fuzzy）的搜尋能力。

3.問題解決

本發明翻譯工具，先天上和語言獨立無關的特性，可以解決上述的問題。它對於新的文字片段，所做的差別

五、發明說明（７）

式的重要性衡量(differential weighting)，提供針對文字、片語以及單及多句子文件「模糊匹配」(fuzzy match)的能力。對於句子中的子字串，「模糊匹配」提供有效的「模糊」索引搜尋能力。

發明概述

本發明關於一種供電腦輔助翻譯之「翻譯記憶庫」(translation memory)。數個原始語言文字字串在與目標語言的文字字串配對後，依序的以可供電腦讀取的格式，被儲存成位於電腦可用記憶體中的一個「循序檔案」(aligned file)。此一「循序檔案」中每一原始語言文字字串的「位置向量」(posting vector)，也會被存放在電腦可用記憶體中的一個「位置向量檔案」(posting vector file)中。每一個位置向量包含一個文件識別號碼以及數個相關重要度(entropy weight)。此文件識別號碼會對應到上述「循序檔案」中的一個原始語言文字字串，而相關重要度(entropy wight)會對應到所選的原始語言文字字串中的一個獨特的n-gram字母。翻譯記憶庫最好還能包含一個「顛倒的索引」，包括有：一個原始語言字母n-grams的清單、每個列出的字母n-gram的相關重要度(entropy weight)以及位置向量的數目（包含一個所列出字母n-gram的項目、以及一個指到每一位置向量的指標，該指標又包含一個所列出的字母n-gram的項目）。

從另一個角度來看，本發明亦代表一個建立「翻譯記

五、發明說明（8）

憶庫」(translation memory)的方式，它並有一個衡量過的字母n-gram檔案給原始語言。一個具有多種原始語言文字字串的「循序檔案」，每一原始語言文字字串並都與一目標語言文字字串配對。每個原始語言文件都會有一個「文字片段向量)(text segment vector)。該向量包含有一份同時出現在原始語言文件，以及上述衡量過的字母n-gram檔案中的字母n-gram之清單。在文字片段向量中的每個字母n-gram，與來自原始語言的衡量過的字母n-gram檔案中相對的n-gram之相關重要度(entropy weight)相關連。每個文字片段向量的相關重要度(entropy weight)最好能「一般化」(normalized)，以反映出每一原始語言文件的長度。這些文字片段向量最好能產生出「顛倒的索引」。此一「顛倒的索引」包含一份出現在文字片段向量中，獨特字母 n-grams 的清單。每份針對「顛倒的索引」之獨特的字母n-gram清單，具有一組識別碼，能夠指到包含所關連的字母n-gram的「循序檔案」中的原始語言文字片段。

本發明同時亦提供特定的方法，以及電腦軟體工具，以達成語言的分析。其做法為：使用字母n-grams、利用字母n-grams為以前翻譯過的文字建立索引、以及從一個翻譯記憶庫利用字母n-grams找回（所搜尋的）文字。

圖式簡述

圖1顯示根據本發明之語言分析方法各步驟流程的說明；

五、發明說明（9）

圖2顯示根據本發明在處理entropy（相關性）「一般化」（normalization）各步驟流程的說明；

圖3顯示根據本發明之「顛倒的索引」（inverted indexing）方式各步驟流程的說明；

圖4顯示根據本發明之翻譯記憶庫檔案結構的說明；

圖5顯示根據一較佳實施例之一個成對的「循序檔案」之檔案格式；

圖6顯示根據一較佳實施例之一個位置向量檔案之檔案格式；

圖7顯示根據一較佳實施例之一個「相關檔案」（correlation file）之檔案格式；

圖8顯示根據一較佳實施例之一個「顛倒的索引」（inverted index）之檔案格式；以及

圖9顯示根據本發明之文字（搜尋）找回方法各步驟流程的說明。

圖式詳細說明

1.概述

一種翻譯記憶庫TM提供一個能夠快速的找回以前已經翻譯過的文字之裝置，翻譯記憶庫的一個主要目標是「模糊」（fuzzy）或是「近似」（approximate）的匹配。「模糊匹配」的搜尋方式可以找出文字順序、形態、大小寫以及拼法稍有不同的句子。「自然語言」文字中，大量的變化之可能性，使得「近似」匹配的搜尋方式也是需要的。

五、發明說明（10）

根據本發明之翻譯記憶庫，使用一個以衡量過的字母n-grams為基礎的架構。這裡用到的"n-grams"、"letter n-grams "或是"字母n-grams"代表的是包含有n個連續字母的文字字串。找回文字的工作，是藉著將翻譯記憶庫中的文字片段（即字、片語、或是句子），以衡量過的n-grams向量（vectors of weighted n-grams）表示，而達成的。向量會經一個合適的相似性函數比較，例如是向量的餘弦函數。這個相似性函數會產生一個分數，以用來評鑒匹配的相似性，所以最相似的文字片段會出現在清單的最前面。一個叫做vector-based retrieval（以向量為基礎的搜尋找回）的技術被用來提高搜尋匹配程序的速度。此一vector-based retrieval模型，是達到快速sparse（稀疏）之向量計算的一個技術。

根據本發明的翻譯記憶庫TM之核心，是一個「循序檔案」（aligned file）。它包含一個分隔成多個文字片段的原始語言檔案。一個文字片段可能是一個字、一組字、一個片語、或是一個句子。每個原始語言的文字片段都會被連繫或對應到一個已經翻譯好的目標語言文字片段。許多下面所述的運算，僅須用在原始語言檔案中的文字片段。然而在本文的討論中，不要忘記了每個原始語言文字片段，在「循序檔案」中，都有一個對應且已經翻譯好的文字片段。所以在搜尋原始語言文字片段的同時，亦會找出已經翻譯好的（目標語言）文字片段。

本發明的翻譯記憶庫TM最好能以乾體的形式，在可供

五、發明說明（11）

一般目的使用電腦上執行。本發明中的翻譯記憶庫及方法的軟體，已經在IBM-PC及與其相容的個人電腦上發展出。本發明中的翻譯記憶庫及方法亦可應用在其它電腦硬體系統上。

下面四個主要模組，可以說是根據本發明之翻譯記憶庫最簡單的說明：

1.語言分析模組(Language Analyzer Module)

2.相關重要度(entropy weight)的「一般化」(Normalization)

3.索引(Indexing)

4.找回文字模組(Retriever)

下面會個別的討論這些模組。至於根據本發明的翻譯記憶庫方法之發展及使用，另有詳細的討論。

語言分析模組(Language Analyzer Module)

語言分析模組的目的是根據任一文字，確定一個獨特的字母n-grams，並衡量每個n-gram，以提供其相對的重要性（相關性）。於此較佳實施例中，「語言分析模組」一開始會確定所有獨特的字母n-gram，而不考慮內容或是出現的次數。

「衡量」(weighting)的目的是要自動的排除掉"嘈雜"(noisy)的n-grams(即不重要的n-grams)。嘈雜的n-grams包括字尾(suffixes)、接語(affixes)以及短並且常出現的字（例如英文中的the、to、of之類的字）。因為它們時常出現的關係，這些n-grams並不會區分文字字串

五、發明說明（12）

。相對的，字根（word roots）通常是由較少出現的n-grams所組成的。所以「衡量」（weighting）可以排除掉比較沒有意義的n-grams，而只留下比較有意義的n-grams。

此「語言分析模組」可以用來分析出任何數目的文字樣本，以針對某一特定語言，建立一個相關n-grams的資料庫或是歷史檔案。分析大量原始（語言）文字，應該可以提供針對該語言一個很有用處的重要n-grams之索引。就像是一個傳統的字典一樣，n-grams提供該語言一個特徵。與傳統字典不同的是，「語言分析模組」所斷定出較重要的n-grams，代表該語言更基礎的特性，因為這些n-grams與任一語言的習性獨立無關。

圖1所示的步驟101中將要用來分析的文字範例，編碼成電腦可讀的格式。鑑於現有可供電腦讀取的文字非常的多，步驟101的目的可能已經被達到了，而不需任何額外的編碼。每一種語言均以一個特定（且通常是統一）的編碼集（codeset），編碼成電腦可讀的格式。例如美國常用ASCII或任一由其延伸出的編碼集來為文字編碼。歐洲地區的電腦用戶通常使用Latin-1編碼集。日文通常使用JIS編碼。這些編碼集基本上是不相容的，但是本發明利用下述方法彌補此一不相容的問題。

這裡的文字範例可為一個要被翻譯的原始語言檔案，一個已經被翻譯過的原始語言檔案，或是一個僅供原始語言參考的範例之原始語言檔案。每個原始語言檔案包

五、發明說明（13）

含許多例如是字、片語、句子或是段落之類的文字片段。如何分割原始語言檔案內的內容，基本上是一個設計上的決定。針對大多數翻譯的目的，一般認為將原始語言檔案分割成代表句子的文字片段是最有用處的。

步驟103會讀取原始語言檔案的文字編碼集和位置的資訊。這些資訊可能結合在文字範例中，或是可以透過文字範例以人工或是自動的方式取得。必須先取得這些資訊，才能適當的將範例文字檔案解碼。

自步驟109，文字範例中的每個文字片段會被選擇，並依序在步驟109至117中被處理。在步驟109中，所選擇的文字片段會先被「分成小塊」(tokenized)。此一步驟109會產生出一組包含在所選擇的文字片段中的字母n-grams。於一較佳實施例中，英文以及印歐語系的語言使用trigram(即3-gram，三個連續的字母)，而韓文、日文以及中文之類的亞洲語言使用bigram(即2-gram，兩個連續的字母)。這裡必須具體指出的是，本發明對n-gram的大小並沒有限制。任何大小的n-gram均可選擇，包括1-grams、2-grams、3-grams、4-grams、5-grams、6-grams或是更高的n-grams。在某些應用上，不同的n-gram大小會很有用處。一個使用多種n-gram大小的單一翻譯記憶庫也是在期待中的。N-grams的選擇，亦可依據原始語言中的音節(數目)。

「分成小塊」(tokenizing)的步驟，使用重疊的n-grams。

五、發明說明（14）

例如下面的句子：

The boy ran.

可以分割成下面的trigram:

Th; The; he; e_b; _bo; boy; oy-; y-r; _ra; ran; an-

上面的"_"，代表的是字母之間的一個空白。在真正的應用上，會使用真的空白字母，這裡的"_"僅用來幫助說明瞭解。每個trigram，在下文中會被稱為一個「小塊」(token)或是字母n-gram。

於一較佳實施例中，步驟111會將「小塊」轉換成「單碼」(Unicode)。雖然本發明並不強制要求此一「單一碼」的轉換，這種轉換可以提供依據本發明所發展出的系統，對於語言更佳的獨立性。單一碼是一個16位元(bit)的編碼集，以供所有現在有在使用的語言，以及一些已停止使用的語言編碼用的。每個單一碼的字元，由16個位元來代表，所以此單一碼編碼集一共可以包含65,000個獨特的字元。作為一個比較，ASCII 編碼集中，每個字元僅需7個位元。步驟111中單一碼的轉換，提高了對記憶體的要求，但是它使得下面的處理模組能夠獨立於所處理的語言之外，因此提供本發明一個很重要的優勢。由步驟111單一碼轉換所產生的結果是每個n-gram均由一個獨特的單一碼來代表。所以所選擇的文字序列（片段）會以該文字序列中，每個n-gram的單一碼序表示出。

五、發明說明（15）

在所選擇的文字序列中，每個n-gram的（出現）次數會在步驟113中列成一表。在上面的例子中，每個出現的n-gram僅在該簡單的句子中出現一次，但是在比較複雜的句子中，n-gram可能會出現多次。步驟113列表的結果，是一個近似於下面表1，根據每個所選擇文字片段，一組成對的"次數與n-gram"的資訊（frequency:n-gram）。

_Th	The	he_	e_b	_bo	boy	oy_	y_r	_ra	ran	an_
1	1	1	1	1	1	1	1	1	1	1

表1

這組一對對"次數與n-gram"的資訊（frequency:n-gram）會在步驟115中，被儲存到一個例如是磁碟機的儲存媒體上，以供稍後使用。

每個獨特的n-gram出現在整個範例文字檔案中，累積的次數，也必須記下來。此一需求可以輕易的在步驟117中達到。該步驟會將步驟113的（次數）表中的次數（frequency）加到一個整體性（global）的n-gram出現次數檔案。此一檔案和步驟113所說明的成對之"次數與n-gram"的資訊（frequency:n-gram）相似。但是它包括在（整個）範例文字檔案中出的n-grams（次數），因此這裡的數目可能會達到數千之多。同時，許多n-gram:frequency檔案中的n-grams，會出現多次。在英文裡，常見的n-gram（例如是"_i_"），可能會在任一原始文字

五、發明說明（ 16 ）

檔案中出現數百次。

上述的整體性n-gram:frequency檔案，一旦在為一個選擇的文字片段被更新過後，流程會回到步驟107，使得每個被選擇的文字片段，依序的通過步驟109到步驟117。一旦當時所選的範例檔案中，所有的文字片段都已經被處理過後，流程會回到步驟105。類似於步驟107，步驟105會重複步驟107至步驟117，以處理每個需要處理的範例文字檔案。

在所有的原始檔案都被處理過後，上述的各步驟亦同時產生了兩個重要的檔案。第一個檔案的結構是根據各別文字片段所建立的，它並包括各文字片段的n-gram:frequency資訊。第二個檔案是整體性(global)的n-gram:frequency檔案，它包括一個可達數千個n-grams的表，以及各n-gram出現在範例文字檔案的次數。

步驟119會為每個n-gram計算出一個相關重要度。相關重要度(entropy weight)的目的，是在顯示出任一特定的n-gram，在原始語言中所有出現之n-gram上下文的相關性。例如從數學的角度來看，英文（英文有27個基的本字元）最多可有27^3或是19,683個trigram(即3-gram)。如果加上大小寫的變化，以及其它常用的字元(符號)，此一數字會大大的提高。但是因為音韻上的限制，實際的數字要比上面所述少得很多。語言分析發現僅有數千個n-grams才有足夠的出現次數，以被考慮成是相關的。

五、發明說明（17）

下面是一個衡量相關重要度(entropy weight)的方程式：

$$Entropy_i = 1 - \frac{\sum_{k=1}^{N} \frac{freq_k}{tfreq_i} \log_2 \frac{tfreq_i}{freq_{ik}}}{\log_2 N}$$

上面方程式中的：

$Entropy_i$ =字母 n-gram i的相關重要度(entropy weight)

$freq_i$ =在文字片段 k 中字母 n-gram i的出現頻率

$tfreq_i$ =在所有文字片段中字母 n-gram i的總出現頻率

N=所有文字片段的總數

所有上面方程式所用到的數值，都可以在上述的幾個檔案中找到。所以步驟119會依序選擇每一對n-gram: frequency的資訊，並計算出其相關重要度(entropy weight)。在步驟119計算出相關重要度後，其結果會被歸納成一個類似下表的表格，其中包括i個範例文字檔案中的n-gram，以及它們相對的相關重要度。

n-gram 1	n-gram 2	n-gram 3	. . .	n-gram i-1	n-gram i
1	.44	.29		.67	.21

表 2

並不是每一個n-gram都會有足夠的相關性，以供翻譯記憶庫使用。根據上面的entropy方程式，所有的相關重要度(entropy weight)，都會介於0.0和1.0之範圍間。在步驟121中有一個最低標準過濾的手續，以剔除不相關以及沒有太大用處的n-grams，而不將它們記

五、發明說明（18）

錄在整體性的n-gram:weight檔案中。此一標準有一上限和一下限，如果任何n-gram的值不在上下限之間，它不會被記錄起來。上下限通常是設在0.30和0.99之間，針對任一特定的應用，此上下限可以調整，以達最佳效果。通常重要度（weight）較低的n-grams，都是字尾（suffixes）、接語（affixes）或是例如是"an"及"the"之類常見的字。如果一個n-grams的相關重要度（entropy weight）接近1.0，它極少出現在範例文字檔案中，所以不具原始語言的特色。對翻譯來說，它沒有太大的幫胯。

在步驟123中，過濾的n-gram:weight檔案會被儲存起來，以供稍後的參考。此過濾過的n-gram:weight檔案，就好像是一個字典一樣，列有針對某一語言相關的n-grams，以及一個定義出各n-gram相關程度的重要度（weight）。根據範例文字的性質，此檔案可以任意的被多次使用。下文會印證本發明很多處理過程中，許多不同類型的檔案，都會用到語言分析（模組）。

相關重要度（entropy weight）之「一般化」（Normalization）

從有助於翻譯記憶庫的語言分析程式中，延伸出的是一個「文字片段向量檔案」（text segment vector file）。該檔案包括一個原始檔案中所有的文字片段，相對於各片段之一組獨特的n-grams，以及各n-gram相對的相關性重要度（weight）。這樣的檔案會有下面的一般格式：

五、發明說明（19）

文字片段1

n-gram weight

n-gram weight

n-gram weight

n-gram weight

文字片段2

n-gram weight

n-gram weight

n-gram weight

n-gram weight

、

、

文字片段N

n-gram weight

n-gram weight

n-gram weight

n-gram weight

表3

針對一個具有N 個文字片段的原始文字檔案。這裡應該注意到的是，表3 僅是一個簡化的說明，並不代表一個真正的檔案結構。它僅被用來顯示一個文字片段向量檔案的內容。每個文字片段向量包含數個n-gramS此

五、發明說明（ 10 ）

n-gram的個數可以被當做是文字片段向量的次元。每個次元有一個由其所附的重要度（weight）值所定義的大小。根據文字片段的長度，以及其包含的n-gram內容，每個文字片段向量可以包括任意數目的n-grams：weight項目。

圖2顯示出從一個原始語言文字，建立文字片段向量檔案的步驟。原始語言文字最好是從一個具有多個原始語言文字片段，並與目標語言文字片段配對的「循序檔案」中取得。但是圖2所示的步驟，僅會處理原始語言文字片段。

圖2的處理是由步驟201載入原始語言編碼集（例如是用於英文的ASCII 碼）以及有關位置之資訊開始的。位置資訊提供例如是美式英文或英式英文的進一步說明。在步驟203中，單一碼（Unicode）編碼集被載入，以提供標準單一碼編碼集的資訊。在語言分析過程中所產生之整體性，已經被過濾過的n-gram：wight檔案，會在步驟205中被載入。而且此檔案中每一個獨特的n-gram會在步驟207中被指定一個獨特的識別碼。指定此識別碼之目的是要提供一個稍微簡單一點的方式，以參考上述檔案中數千個n-gram，藉而減輕處理的負擔。

在步驟213和214中，每個「循序對」（aligned pair）會從「循序檔案」中被依序選擇，並在步驟213中被讀取。步驟214會計算所選擇的「循序對」中，每個文字片段的大小（位元組），並依此將它重新格式化，然後再

五、發明說明（ㄙ）

將此重新格式化後的「循序對」及其位元組長度存回。步驟213及214減輕以下處理步驟對「循序對」讀取、搜尋以及找回的負擔。雖然實行步驟213和214是可選擇性的，但是它們大大的提高本發明的翻譯記憶庫以及方式運作時的速度。

步驟211接著處理步驟213及214「循序檔案」中的每一個「循序對」。當有一個以上的「循序檔案」需要處理時，步驟209會同樣地接著處理每一個「循序檔案」。步驟209及步驟211提高在理想的應用下之用途，但是如有需要，它們亦可針對某一特定應用加以修改，或是根本不使用。

「文字片段向量」是蓄意的用來提供針對一個「詢問」(query)所搜尋原始文字檔案的基礎。此一處理過程在下面「找回文字(模組)」一節會有詳細的說明。在此很重要的一點是要瞭解到列在每個文字片段向量中的n-grams及重要度(weight)必須被選擇，以達到有效率之「絕對」(identical)或是「模糊」(fuzzy)的匹配。

如同在「語言分析模組」中所說的，所列出的n-gram會經過過濾，以剔除嘈雜(noisy)的n-grams。其做法是將沒有出現在由「語言分析模組」所產生經過濾的整體性n-gram:weight檔案中的n-grams剔除掉。因為這種嘈雜的n-grams，不但對詢問(query)所作出的配對(matching)沒有幫助，可能還會造成妨礙，所以它們會被剔除掉。前述的步驟115中，儲存有一個n-gram:

五、發明說明（22）

frequency(次數)對的檔案，其中的原始語言文字片段已經被「語言分析模組」處理過了。此檔案可以提供所需要的已「分成小塊」(tokenized)之原始文字，以讓流程得以在步驟222中做「一般化」(normalization)的工作。在步驟217和219中，每個「循序檔案」中的「循序對」(aligned pair)會依次的被選擇。依據上述的方法，步驟221會將所選擇的「循序對」之原始文字「分成小塊」。

原始語言文字檔案中的文字片段之長度，通常可能會有很大的差異，其內容可能是字和片語或是完整的句子及段落。一般來說在一個詢問(query)中，較長的文字片段會包含相同的字母n-grams。因為文字片段越長時，它會有較多的n-grams，而出現相同重複n-gram的機率也就越大。重要度(weight)的「一般化」(normalization)之優點是在避免較長的文字片段所造成的偏差。一個可行的做法是為每對n-gram:sengment的資料，提供一個依其片段長度「一般化」之後的重要度(weight)，以產生一個segment:n-gram:weight的tuple(即一組含三個項目的資料)。此一作法亦可簡化下面「找回文字模組)(Retriever)中所用的相似性占算，因為這裡向量的乘積(經過「一般化」的weight)，和其它更複雜的餘弦計算(cosine measure)產生相同的結果。

下面是一個可用來將文字片段向量的重要度(weight)

五、發明說明（ 2 ）

「一般化」之範例方程式：

$$NormalizedEntropy_{ik} = \frac{(freq_{ik})(Entropy_i)}{\sqrt{\sum_{i=0}^{n}(freq_{ik})^2(Entropy_i)^2}}$$

上面的：

$Entropy_i$ =來自整體性entropy計算中字母n-gram i的相關重要度（entroy weitht）

feq_{ik} =在文字片段k中，字母n-gram i出現的頻率

n=所有獨特字母n-grams的總數

步驟222會為原始語言檔案中的每個文字片段之每個n-gram，個別地計算「一般化」的entropy值。這種處理的結果，造成任一n-gram在不同的文字片段中，可能會有不同的重要度（weight）。

為原始文字檔案中每個文字片段所產生的文字片段向量，包含一個相關n-grams的清單，其中的每個n-gram都會有一個「一般化」的相關重要度（entropy weight）。前面表1所討論的文字片段，在此可能會變成這樣：

he_	e_b	_bo	boy	oy_	_ra	ran
.35	.47	.55	.7	.31	.32	.57

表 4

因為嘈雜（noisy）的n-grams已被過濾掉了，一些出現在表1中的項目因此被剔除，而沒有出現在表4中。同時需注意的是，表4中的重要度（weight），已經經過上述的方法「一般化」過了。表4所代表的是每個文字片

五、發明說明（ 14 ）

段向量所含有的資料，並可以用任何合適的方法，儲存在資料庫檔案中。下文中會說明一個理想的檔案組織結構，以提供良好的搜尋能力。

4.索引（Indexing）

索引（indexing）對於從翻譯記憶庫中快速的找回資訊是很重要的。因為翻譯記憶庫的目的是要當做翻譯人員的輔助工具，它必須能夠即時的（real time）答覆翻譯人員的詢問。達到即時答覆的原理是建立一個有相關性的n-grams表（不含有嘈雜的n-grams），並附上所有包含這些n-gram的文字片段。此表稱為「顛倒的索引」（inverted index），它可以快速的找出翻譯人員所有興趣的（文字）片段，而忽略掉其它的片段。「顛倒的索引」需要額外的電腦記憶體和磁體空間以供儲存，其大小可能接近甚至超過「循序檔案」（aligned file）的大小。

本發明之翻譯記憶庫系統較佳的索引步驟（indexing algorithm）是由美國維吉尼亞理工學院（Virginia Tech.）的Edward A. Fox和Whay C. Lee之FAST-INV系統所演變出的。FAST-INV的原理是將text segment:n-gram:weight資料組（tuple）分散成（較小的）「部分」（load），以讓它們可以在主記憶體中處理，因而避免掉其它必須讀寫磁碟機很慢的排序（sorting）之需要。所以本發明可以使用一般具有8Mb 主記憶體的標準式個人電腦，為數十億位元組（gigabytes）的文字建立索引。雖然上述在理想應用下的索引方式可以提供優良的性能，亦可

五、發明說明（ 25 ）

使用一些其它有名的文字找回設計(text retieval schemes)，不過它們的性能會稍微差一點。常見的例子包括：搜尋樹狀結構(search tree structures)、拼湊(hashing)功能以及數位樹狀(digital trees)。此外，其它找回文字的方式亦會包括循序的檢視儲存在記憶體中的文字，以及許多其它顛倒式的索引結構，包括排序過的矩陣(sorted arrays)、B-樹狀(B-trees)以及試作(tries)等。在本發明理想的應用中所說明的特定之索引設計(indexing scheme)並不是本發明的一個限制。

如圖3中所示之較佳實施例中之「顛倒的索引」(inverted index)之建立。步驟301會從在「語言分析模組」所建立的經過過濾之整體性n-gram:weight檔案中，載入一份獨特的n-grams以及重要度(weights)到該步驟中。類似於前面步驟207的做法，每個n-gram會被指定一個參考號碼(reference number)。步驟303會建立一個「部分表」(load table)。該步驟會先檢視所有的segment:n-gram:weight資料組(tuple)，並計算出「部分」的個數，以使得所有「部份」均可被容納在可用的記憶體中。此一「部份表」包括從零開始，到最大識別碼(ID number)為止，數個範圍的n-gram之識別號碼。例如，第一個「部份」可能僅包括0至214號的n-grams，而第二個「部分」包括215至302號的n-grams。估計「部分」的範圍之方法是，當僅要處理該範圍內的n-grams時，計算出有多少對的n-gram:

五、發明說明（ 26 ）

weight資料，可以被安全的容納在可用的記憶體中。這裡建立的「部分表」，是FAST-INV步驟原理的一個特殊功能，它使得僅有有限記憶體的個人電腦，得以為具有數十億位元組(gigabyte)文字的檔案建立索引。

一旦步驟303建立了「部分表」(load table)之後，每個「部分」會依序的通過步驟307至步驟311之處理。記憶體中會建立有一個「顛倒的索引」。在圖2中所述的「一般化」過程中，所建立的文字片段向量內，所有的text segment:n-gram:weight資料組(tuple)，在此會依序的被讀入。如果一個文字片段包含一個在當時的「部分表」之中的n-gram，該片段的識別（號碼）會被加到步驟309中之「顛倒的索引」內相對的項目。然後在步驟311中，此「顛倒的索引」會被儲存到一個例如是磁碟機的永久性（儲存）記憶體。

所儲存的「顛倒的索引」檔案包括每個n-gram的一個指標。該指標會指到位置向量(posting vector)檔案中，一個獨特的位置向量。索引中的每個n-gram，在位置向量檔案中會中有一個位置向量。每個位置向量包含一份文件識別（碼）的清單。針對所選擇的文件中所選擇之n-gram，每個文件識別（碼）附有經過「一般化」的重要度(weight)。在翻譯記憶庫中，每個位置向量會位於一個獨特的地址，而且「顛倒的索引」會參考這些獨特的地址。其達成的方式是在步驟309中，將相對於位置向量的獨特地址加到索引檔中。在「顛倒的索引」結構中

五、發明說明（ 27 ）

含有位置向量之概念是眾所周知的，進一步的瞭解可以從參考資訊找回文字中得到。

一旦所有的「部分」(load)都被處理過後，步驟305將流程指向步驟313，以組合出翻譯記憶庫。在步驟215中所建立之「循序對」(aligned pair)檔案，會在步驟313中，被複製到翻譯記憶庫。這使得每個「循序對」(aligned pair)在翻譯記憶庫TM中，指到一個獨特的地址。步驟315會將位置向量，以及已完整之「顛倒的索引」，複製到翻譯記憶庫TM中。在經過步驟317提供檔案「標題」(header)的資訊後，翻譯記憶庫TM即建立完成。檔案「標題」資訊包括有用的細節，例如是使用的n-gram之大小、「顛倒的索引」之開始地址、翻譯記憶庫TM中「循序對」(aligned pair)的個數、以及其它有關「循序對」資料結構和位置的詳細資料。針對任一特定應用的需要，其它資訊亦可被複製到檔案「標題」中。

為幫助瞭解，圖4中有一個本發明之翻譯記憶庫TM400的結構圖。除了圖4所顯示出的部分，沒有顯示出的「標題」(header)資訊也會被加到翻譯記憶庫TM400中，以儲存一般的資訊，來說明本翻譯記憶庫TM結構的細節，以及其它上述資料項目的位置。顯示在403的「循序對檔案」(aligned pairs file)之細節可以在圖5中找到。此檔案包括每個「循序對」(aligned pair)，以及其在「循序對檔案」中，每個原始語言文字片段及目標語言文字片段之大小。片段大小的資訊，使得「循序

五、發明說明（>8）

檔案」(aligned file)中一對對的資料可以很快的被依序讀取。

顯示在405的位置向量檔案，包括有索引檔409之中，每個n-gram的位置向量。如圖4中的箭頭所示，每個索引檔409中的n-gram會對應到一個在位置向量檔案405中，位於獨特地址的一個位置向量。圖6是一個位置向量檔405的結構範例。為了簡化並加快FAST-INV之類的步驟原理，對於索引及找回文字的處理，位置向量檔405列出文字片段的識別號碼，而不直接的去參考「循序檔案」403中每個文字片段之獨特地址。如果使用其它索引及找回文字的原理，此一步驟可以被取消。

「相關檔案」(correlation file)407的細節可以在圖7中找到。此檔案是用來儲存相關於位置向量檔405中，具有來自「循序對檔案」(aligned pair file)403的獨特地址之文字片段識別號碼的資訊。藉著使用「相關檔案」407，每個405檔案中的位置向量，因此可以被追溯回「循序對檔案」403中的數個文字片段。

圖8是「顛倒的索引」(inverted index)檔409的詳細說明。此檔案可以被想像成一個列有針對原始語言之多個獨特的n-grams 的表。這裡真正列出的n-grams，是由前述之「語言分析模組」所決定的。每個n-grams亦附有它在步驟119中所決定的相關重要度(entropy weight)。這裡列出的n-grams及重要度(weight)，是來自圖1中的步驟123所儲存的經過過濾之整體性n-gram

五、發明說明（ 29 ）

:weight檔案。除了相關重要度(entropy weight)，每個n-gram項目亦有一個包含有此n-gram項目的文字片段數目。此一數字描述在檔案405中，相對的位置向量之大小。此索引檔並包括一份指標的清單。針對每個n-gram，這些指標會指到其對應的位置向量之獨特地址。在一個特定的範例中，這個索引檔被儲存成一個拼湊(hash)表。所以在「顛倒的索引」中搜尋一個特定的n-gram時，所有「循序對檔案」(aligned pair file)中，包含有該特定n-gram的文字片段，都會被找出。

5.找回文字模組(Retriever)

此一「找回文字模組」利用翻譯記憶庫，從「循序對檔案」(aligned pair file)401中，快速的找出類似所詢問(query)的(文字)片段之所有文字片段。所有俱有與詢問片段之n-grams相同的文字片段都會被給予一個分數。此評分的基礎來自與該片段儲存在一起經過「一般化」的重要度(weight)。俱有相符的n-gram之片段會被提供給使用者(翻譯人員)，其順序並會依據其所被賦予的相關性分數排列，最相關的片段會先出現。每個列出的片段都會有一個介於100(完全相關)與0.0(不相關)之間的分數。在一個理想的應用下，如果一個文字片段產生80%以下的「模糊」(fuzzy)相關分數，它不會被提供給使用者。但是此一標準，可以針對任一特定應用之需要加以調整。

圖9是此一「找回文字」(retriever)過程之各步驟

五、發明說明（20）

。本發明之「找回文字」的部分，假設一個依據上述手續所建立的合適之翻譯記憶庫，或是一個與其相當的資料庫已經被建立好了。步驟901會載入一個翻譯記憶庫檔案，而使用者在步驟903鍵入一個所詢問（query）的片段。在步驟905及907，此詢問（的片段）會被「分成小塊」（tokenized）、衡量其重要性（weighted）、過濾、以及「一般化」。其方式類似於前面語言分析以及entropy「一般化」模組所述之步驟。在步驟907之後，此詢問會以一個「詢問向量」的方式代表。此向量包含一串沒有被過濾過程剔除掉的，詢問中所有獨特n-gram之「一般化」之後的重要度（weight）。

在步驟909中，每個詢問向量的n-gram會依序被選擇，並被送到步驟911、912以及913處理。每個n-gram會依序被選取，並用來讀取索引檔409。此索引檔409會單獨的，或是配合「相關檔案」（correlation file）407，提供一些指標，以指到位置向量檔405中的一些特定位置。被步驟911指到的位置向量，會在步驟912中被讀到記憶體，並和詢問向量比較，以測試其相關性。很多已知的計算方式都可用來計算兩個多次元向量的邏輯距離或是相似程度。譬如向量的餘弦（cosine）函數即是一個例子。

於此較佳實施例中，使用「一般化」的向量，可以簡化決定詢問向量與翻譯記憶庫之文字片段向量之間之相似性的步驟。步驟913會建立一個矩陣。每個「循序對

五、發明說明（21）

檔案」（aligned pair file）403中的文字片段都會出現在此矩陣中。每個矩陣項目都會有一個分數，而且這些分數在開始時會被預設成零。當每個n-gram依序被選擇時，在詢問向量中被選擇的n-gram之「一般化」的相關重要度（entropy weight），會和「循序對」（aligned pair）中每個文字片段中被選擇的n-gram之「一般化」的相關重要度（entropy weight）相乘。如同前面已經說過的，「循序對」（aligned pair）中每個文字片段中被選擇的n-gram之「一般化」的相關重要度（entropy weight），可以從位置向量取得。上述相乘的的乘積，會被加到該文字片段項目的分數中。在詢問向量中的每個n-gram被處理的同時，矩陣會累積出一個分數。於較佳實施例之方法中，其值會介於0.0和1.0。它代表詢問向量和每個文字片段向量相似的程度。

當詢問向量中所有的n-gram都被處理過後，流程會跑到步驟915，將整個矩陣依據分數由高到低的順序（decreasing）排序。於理想之實施例中，矩陣的內容也會被區分（grouped），所有俱有較高分數的文字片段會被放在一起，以如快找回文字的速度。排序和區分矩陣是可選擇性的功能，但是它們會大大的提高本發明之用途。在步驟917中，文字片段會從排序過的矩陣中被取回，所以最相似的文件會優先的被取回。

如果一個詢問向量和一個文字片段向量之間有一個完全的符合（exact match），步驟913中的分數會趨近於

五、發明說明（22）

100。因為文字片段（位置）向量，以及詢問向量中，都不包含任何有關該語言特定之文法（grammar）、語法（syntax）或是文字結構方面的資訊，即使在「循序對」文字（aligned pair text）與詢問的文字之間，有拼錯、少許的差異以及文字順序不同之類的情況下，「找回文字模組」仍可輕鬆的偵測出符合的文字（match）。

本發明之翻譯記憶庫TM的另一優點是它用n-grams，而不是「文字」，當做分析文字的基礎，因此本翻譯記憶庫TM已經經過證明是與語言真正獨立的。這所代表的是本文所述的步驟，可以基本上用相同的處理方式，而與原始文字的語言及位置無關。只要知道原始文字的編碼方式，以將之記錄成「單一碼」（Unicode），本發明在任何語言上均提供相似的性能。

本發明之方式的另一個沒有預料中的優點是它提供非常快速的「模糊」（fuzzy）找回文字之能力，所以此翻譯記憶庫TM可以值得讚賞的被當做是一個電子索引（electronic concordance）。當使用者輸入一個字或是片語當做詢問時，「找回文字模組」會從包含該字或片語的「循序對檔案」（aligned pair file）中，找回所有的文字片段。翻譯人員可以檢視這些找回的文字片段，以瞭解這些字和片語在不同場合的用法及意義。

本文中所提出有關檔案格式的特定範例和細節，並不是本發明之方法的限制。對「找回文字」（text retrieval）領域有經驗的專家，可以很快的找出針對本

五、發明說明（ 33 ）

發明的修改，並且針對任一特定的應用，做出適當的修改。需要明確的瞭解到的是本發明之申請專利範圍，不僅只限於在本實施例之說明，它同時也包括在本發明概念的範疇及精神內，所作出的修改和延伸。

四、中文發明摘要（發明之名稱：機器輔助的翻譯工具）

一種為電腦翻譯所設計之「翻譯記憶庫」（translation memory），其基礎是一具有數個原始語言文字字串，與目標文字字串配對的「循序檔案」（aligned file）。在每個「循序檔案」中的原始語言文字字串，會在一個位置向量檔中有一個相對的位置向量。每個位置向量包括有一個文字識別號碼，以對應到「循序檔案」中的一個所選擇的原始語言文字字串。每個向量並有數個相關重要度（entropy weight），每個值都會對應到出現在所選擇原始語言文字字串中的一個獨特字母n-gram。此翻譯記憶庫最好能再包括一個「顛倒的索引」（inverted index），而該索引又包含一份原始語言字母n-gram表，以及一個指到每個位置向量的指標，包括一個所列出字母n-gram的項目。

A5
B5

四、英文發明摘要（發明之名稱：Machine Assisted Translation Tools ）

A translation memory for computer assisted translation based upon an aligned file having a number of source language text strings paired with target language text strings. A posting vector file includes a posting vector associated with each source language text string in the aligned file. Each posting vector includes a document identification number corresponding to a selected one of the source language text strings in the aligned file and a number of entropy weight values, each of the number of weight values corresponding to a unique letter n-gram that appears in the selected source language text string. Preferably, the translation memory further includes an inverted index comprising a listing of source language letter n-grams and a pointer to each of the posting vectors including an entry for the listed letter n-gram.

六、申請專利範圍

1.一種翻譯記憶庫，包含：

一個可供電腦使用的媒體，具有電腦可讀取的資料，其中電腦可讀取的資料尚包含：

一以電腦可讀取之格式編碼，包含數個原始語言文字片段的「循序檔案」(aligned file)，其中每個原始語言文字片段在該可供電腦使用的媒體中，位於一個獨特的地址，並與一個以電腦可讀取之格式編碼的目標文字片段配對；

一「顛倒的索引」(inverted index)，包含有一串原始語言字母n-gram，其中每個列出的n-gram包括有：一個相關於所列出的字母n-gram之相關重要度(entropy weight)的項目、一個「循序檔案」(aligned file)中包含有所列出之字母n-gram的原始語言文字片段個數、以及一個指到電腦可用記憶體中的一個獨特位置之指標；以及

一具有相關於每個列於「顛倒的索引」中的n-gram之位置向量的位置向量檔案，每個位置向量會位於任一由「顛倒的索引」所指到的獨特位置，且每個位置向量含有：

i)數個文件識別號碼，且每個號碼都會對應到一個在「循序檔案」(aligned file)中之一個所選擇的原始語言文字字串；以及

ii)數個相關重要度(entropy weight)，每個值都會附有一個文件識別號碼。

六、申請專利範圍

2.如申請專利範圍第1項之翻譯記憶庫，其中該電腦可讀取資料尚包括包含有針對原始語言文字字串中每個獨特地址的「相關檔案」(correlation file)，該「相關檔案」中的每個獨特地址由一個文件識別號碼來識別。

3.如申請專利範圍第1項之翻譯記憶庫，其中該字母n-grams具有相當於原始語言2至3個字元(字母)範圍的長度。

4.如申請專利範圍第1項之翻譯記憶庫，其中計算該相關重要度(entropy weight)的方法係：

$$Entropy_i = \frac{1 - \sum_{k=1}^{N} \frac{freq_k}{tfreq_i} \log_2 \frac{tfreq_i}{freq_{ik}}}{\log_2 N}$$

其中上面方程式中的：

$Entropy_i$ =字母n-gram i的相關重要度(entropy weight)

$freq_i$ =在文字片段k中字母n-gram i的出現頻率；

$tfreq_i$ =在所有文字片段中字母n-gram i的總出現頻率；

N=所有文字片段的總數。

5.如申請專利範圍第1項之翻譯記憶庫，其中該電腦可讀取資料係在該電腦可用媒體上經過壓縮處理。

6.如申請專利範圍第1項之翻譯記憶庫，其中該位置向量檔案係利用稀疏(sparse)向量編碼的方式壓縮。

7.如申請專利範圍第1項之翻譯記憶庫，其中各位置向

六、申請專利範圍

量之相關重要度(entropy weight)被一般化(normalized)。

8.如申請專利範圍第1項之翻譯記憶庫，其中所有所列出的字母n-gram均以「單一碼」(Unicode)格式提供。

9.如申請專利範圍第1項之翻譯記憶庫，其中會被列出的字母n-gram都具有一個介於事先決定好範圍的相關(entropy)值。

10.一種用於從原始語言翻譯至目標語言以建立翻譯記憶庫之方法，包括下列步驟：

為原始語言提供一個經過重要性衡量(weighted)的字母n-gram檔案；

提供一個包含多種原始語言文字字串的「循序檔案」(aligned file)，每個原始語言文字字串會與一個目標語言文件配對；以及

從「循序檔案」(aligned file)建立一個「顛倒的索引」(inverted index)，此「顛倒的索引」包括一份出現在「循序檔案」中之獨特的字母n-gram，每份在「顛倒的索引」中所列出之獨特的字母n-gram，會附有一組文件識別(號碼)，以指到包含有該字母n-gram的「循序檔案」中的原始語言文字字串。

11.如申請專利範圍第10項之方法，其中為原始語言提供經過衡量過的字母n-gram之步驟包括下列步驟：

提供一個原始語言文字的數量；

將該數量的原始語言文字「分成小塊」(tokenizing)

六、申請專利範圍

，以識別在該原始語言中，一組獨特的字母n-gram；

為每個原始語言中識別出的獨特字母n-gram計算一個相關重要度(entropy weight)；

過濾此組獨特字母n-gram，以剔除沒有達到事先設定相關重要度(entropy weight)標準的字母n-gram；

過濾此組獨特字母n-gram，以剔除沒有達到事先設定出現頻率標準的字母n-gram；以及

將該組過濾好的字母n-gram，及其各字母n-gram之相關重要度(entropy weight)，儲存到該原始語言經過衡量過的字母n-gram檔案(weighted letter n-gram file)。

12. 如申請專利範圍第11項之方法，尚包括計算步驟前，將「分成小塊」的原始語言文字中，每個獨特的字母n-gram轉換成「單一碼」(Unicode)之步驟。

13. 如申請專利範圍第10項之方法，其中建立一個「顛倒的索引」(inverted index)之步驟包含下列：

依序從「循序檔案」(aligned file)中，選擇每個原始語言文件；

將所選擇原始語言的文件「分成小塊」(tokenizing)，以決定其所包括的各字母n-gram；

過濾此「分成小塊」的文件，以將沒有出現在原始語言經過衡量過的(weighted)n-gram檔案中之字母n-grams剔除掉；以及

在過濾後，將仍保留在所選擇的文件中每個字母

六、申請專利範圍

n-gram，與來自該原始語言經過衡量過的(weighted)字母n-gram檔案中，與其相對應之相關重要度(entropy weight)配對。

14. 如申請專利範圍第11項之方法，尚包括在過濾前，將每個「分成小塊」的文件中之每個n-gram轉換成「單一碼」(Unicode)之步驟。

15. 如申請專利範圍第11項之方法，其中從文件向量建立一個「顛倒的索引」(inverted index)之步驟包括為每個原始語言文件的文件向量執行FAST-INV程式步驟。

16. 一種分析語言之方法，包含下列步驟：

提供數個文字字串，以供分析；

辨別這些文字字串所使用的文字編碼集(codeset)；

依序選擇這些文字字串；

將所選擇的文字片段「分成小塊」(tokenizing)，以決定出一組出現在所選擇的文字文件之字母 n-grams；

將每一組的字母n-grams轉換成(相對的)「單一碼」(Unicode)值；

定義出一組獨特的字母n-grams，其方法是在每個字母n-gram僅在該組獨特的字母n-grams出現一次的情況下，將在「分成小塊」步驟中，所決定的每一字母n-gram，加到該組獨特的字母n-grams；

在這些文字字串中，計算每個字母n-gram，出現在該組獨特字母n-grams的頻率；

六、申請專利範圍

計算出該組獨特字母n-grams中，每個字母n-gram的相關重要度(entropy weight)；

過濾該組獨特字母n-grams，以將沒有達到預先設定之相關重要度(entropy weight)的字母n-grams剔除掉；

將每個仍保留在該組獨特字母n-grams中的字母n-gram，與其所計算出的相關重要度(entropy weight)配對；以及

將這些「字母n-gram:entropy weight)」資料對項目，儲存到一個經過衡量過的(weighted)n-gram檔案。

17. 一種具有多個文字字串的檔案以建立一個顛倒的索引之方法，其步驟包括：

建立一份字母n-grams表，其中所列出的每一個字母n-gram，都有一個超過事先設定標準的相關重要度(entropy weight)；

針對每個列出的字母n-gram，斷定出一組包含該字母n-gram的文字字串。

18. 一種從數個文字字串中找回一個目標子集之文字字串之方法，其步驟包括：

輸入一個文字「詢問」(query)；

將該文字詢問「分成小塊」，以決定一組出現在該文字詢問中的字母n-grams；

過濾該被「分成小塊」的文字詢問，以將沒有達到

六、申請專利範圍

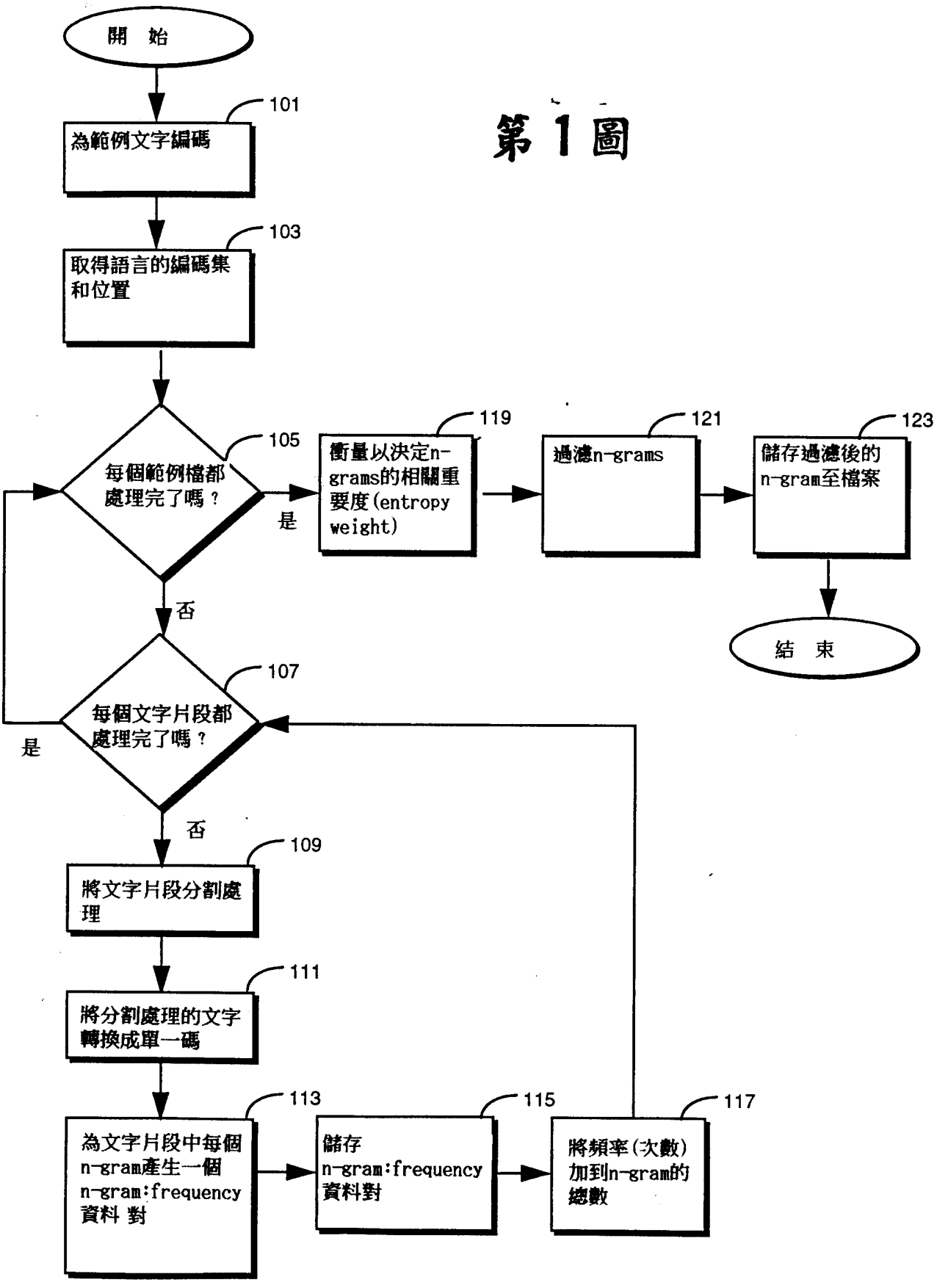
預先設定之相關重要度(entropy weight)的字母n-grams剔除掉；

將每個仍保留在「分成小塊」後的文字詢問中之字母n-gram，與一個相關重要度(entropy weight)配對；以及

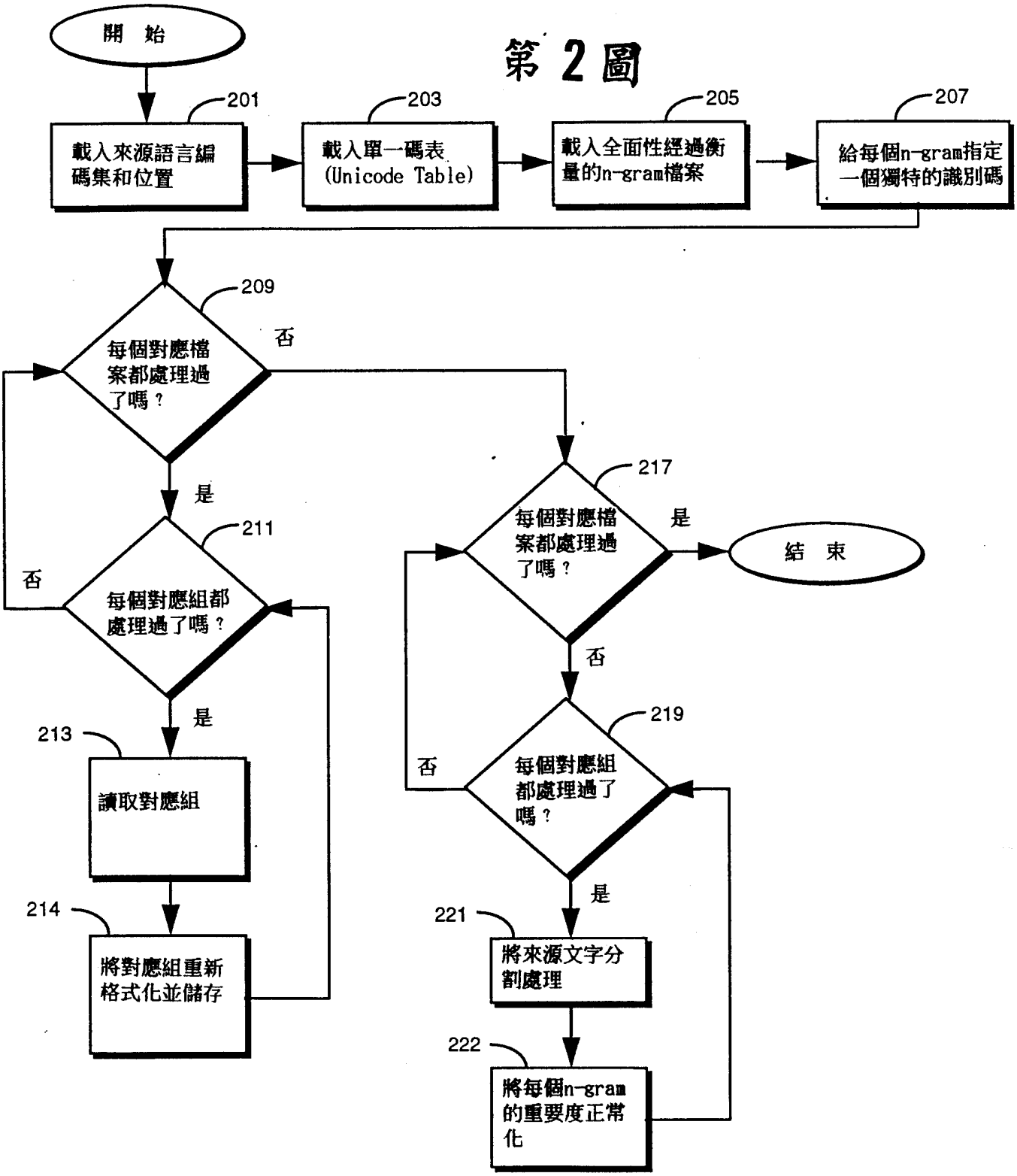
用一個針對那些文字字串之「顛倒的索引」(inverted index)，決定出包含有任何出現的成對之字母n-grams之文字字串的目標子集(target subset)。

85107412

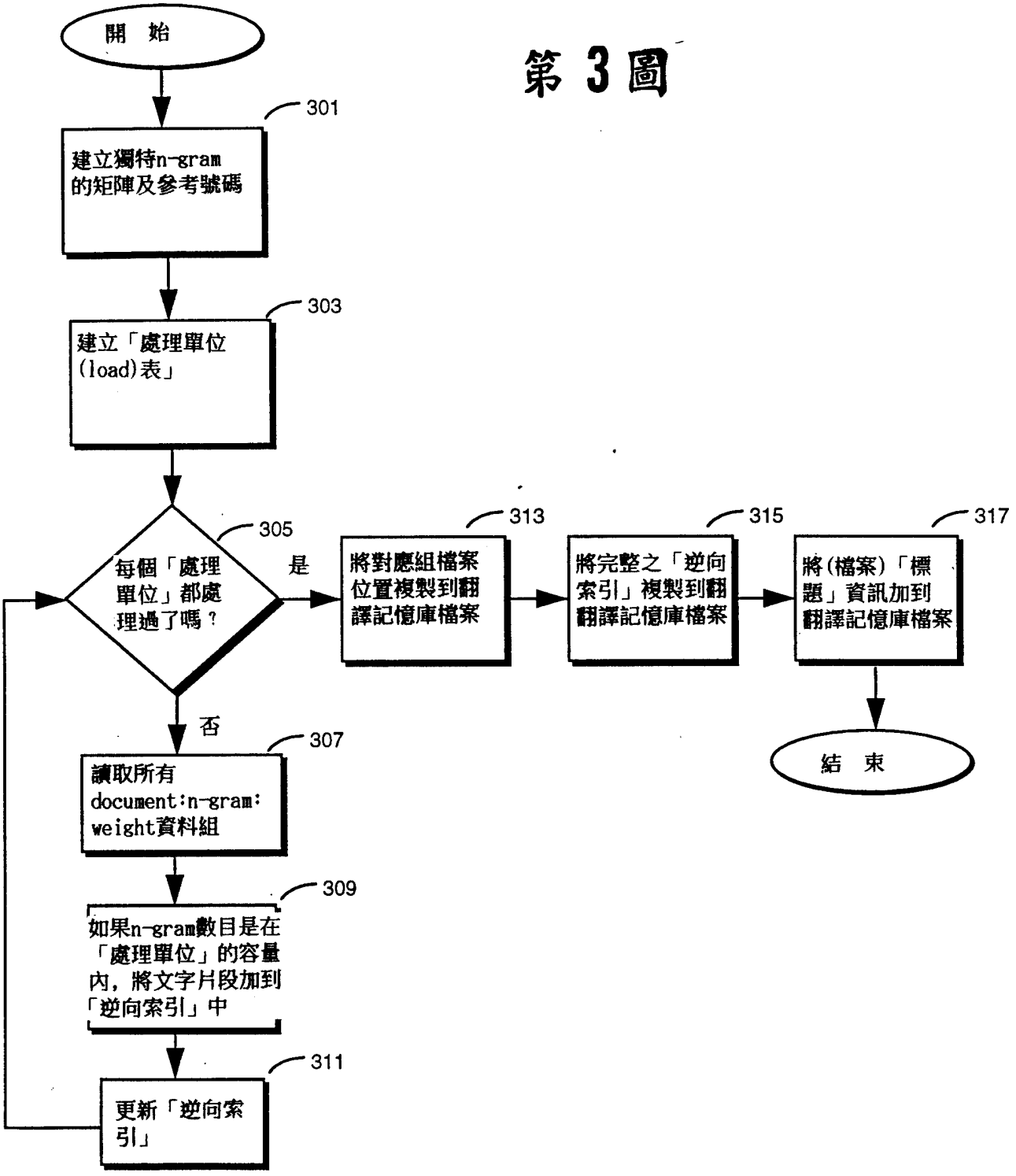
第 1 圖

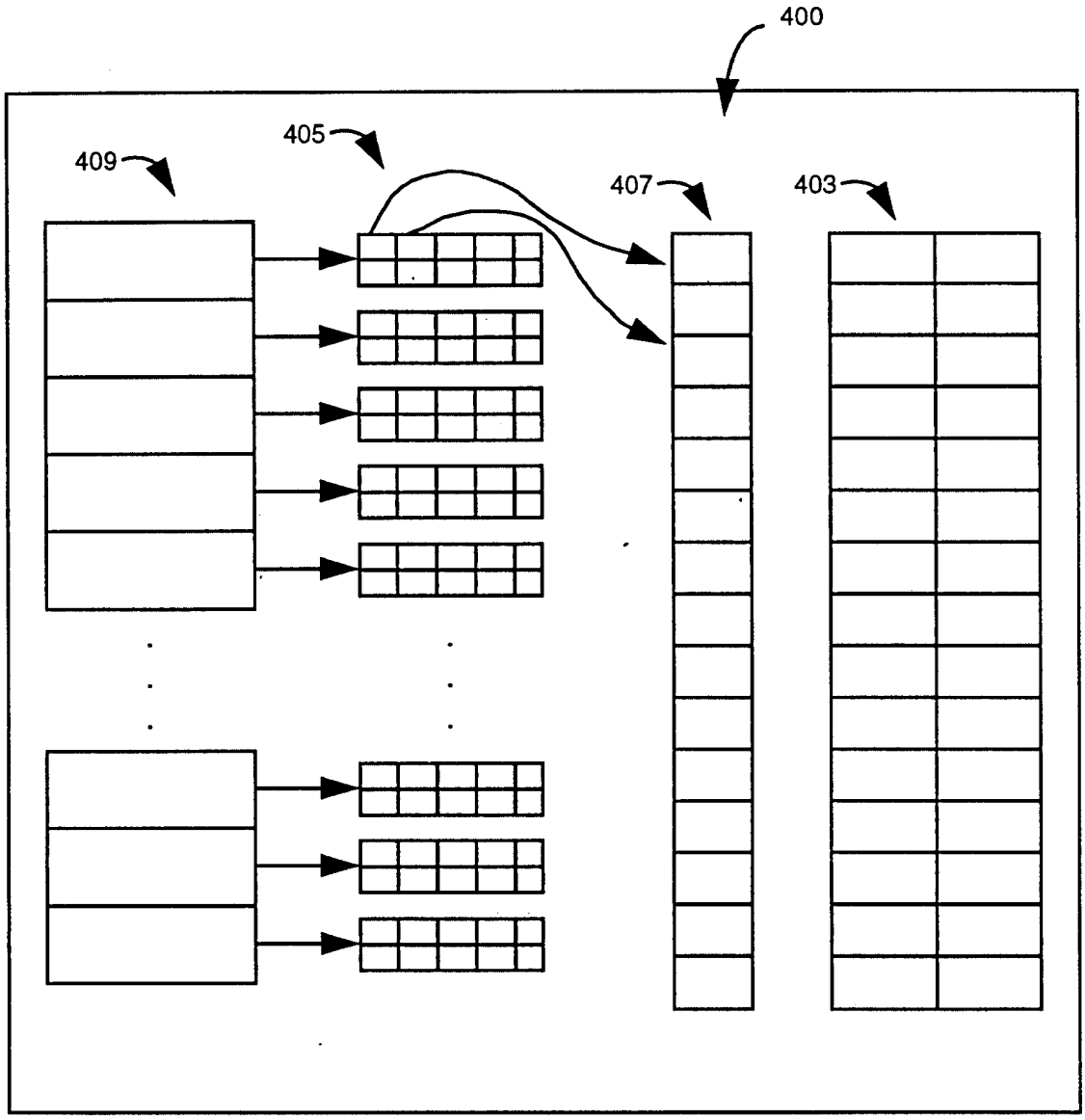


第 2 圖

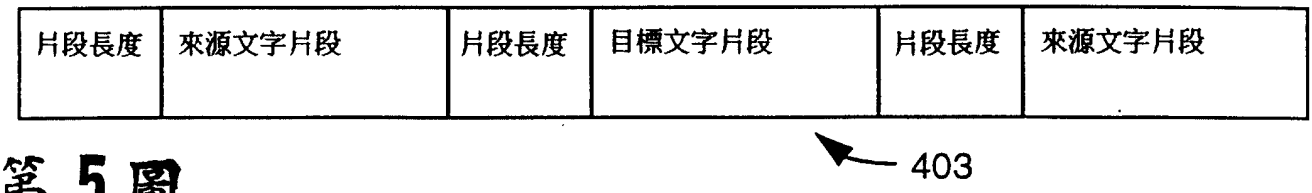

開 始
201
載入來源語言編碼集和位置
203
載入單一碼表 (Unicode Table)
205
載入全面性經過衡量的n-gram檔案
207
給每個n-gram指定一個獨特的識別碼
209
每個對應檔案都處理過了嗎？
否
是
211
每個對應組都處理過了嗎？
否
是
213
讀取對應組
214
將對應組重新格式化並儲存
217
每個對應檔案都處理過了嗎？
是
結 束
否
219
每個對應組都處理過了嗎？
否
是
221
將來源文字分割處理
222
將每個n-gram的重要度正常化

第 3 圖

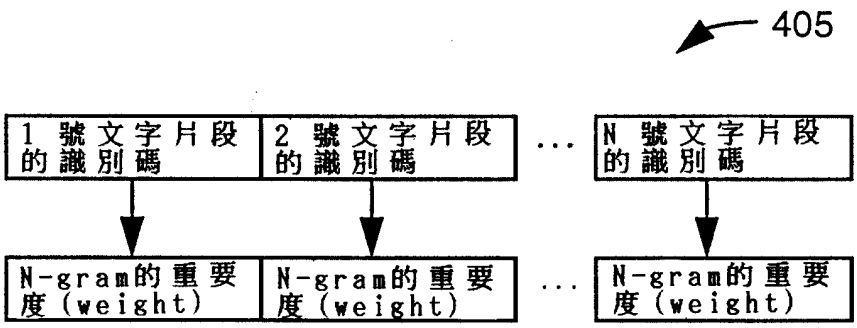

開 始
301
建立獨特n-gram的矩陣及參考號碼
303
建立「處理單位(load)表」
305
每個「處理單位」都處理過了嗎？
是
否
313
將對應組檔案位置複製到翻譯記憶庫檔案
315
將完整之「逆向索引」複製到翻譯記憶庫檔案
317
將(檔案)「標題」資訊加到翻譯記憶庫檔案
結 束
307
讀取所有document:n-gram:weight資料組
309
如果n-gram數目是在「處理單位」的容量內，將文字片段加到「逆向索引」中
311
更新「逆向索引」



第 4 圖


片段長度
來源文字片段
片段長度
目標文字片段
片段長度
來源文字片段
403

第 5 圖


405
1 號文字片段的識別碼
2 號文字片段的識別碼
...
N 號文字片段的識別碼
N-gram的重要度(weight)
N-gram的重要度(weight)
...
N-gram的重要度(weight)

第 6 圖

407

第一個文字片段地址	第二個文字片段地址	第三個文字片段地址	...	第N 個文字片段地址

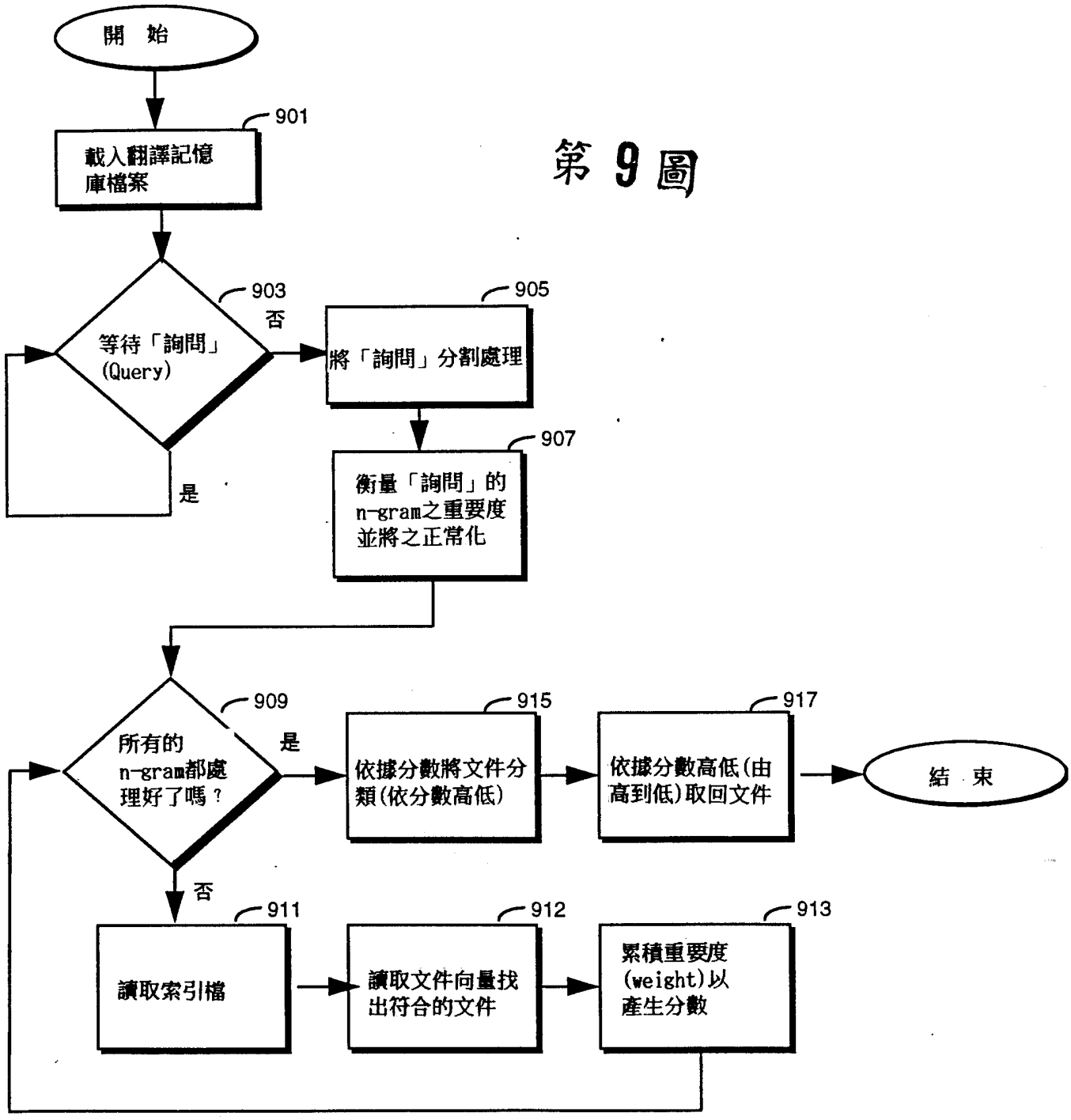
第 7 圖

409

N-gram 1	相關重要度 (entropy weight)	位置向量的個數	指到位置向量的指標
N-gram 2	相關重要度 (entropy weight)	位置向量的個數	指到位置向量的指標
N-gram 3	相關重要度 (entropy weight)	位置向量的個數	指到位置向量的指標
.	.	.	.
.	.	.	.
.	.	.	.
N-gram n	相關重要度 (entropy weight)	位置向量的個數	指到位置向量的指標

第 8 圖

第 9 圖


開 始
901
載入翻譯記憶庫檔案
903
等待「詢問」(Query)
否
是
905
將「詢問」分割處理
907
衡量「詢問」的n-gram之重要度並將之正常化
909
所有的n-gram都處理好了嗎？
是
否
915
依據分數將文件分類(依分數高低)
917
依據分數高低(由高到低)取回文件
結 束
911
讀取索引檔
912
讀取文件向量找出符合的文件
913
累積重要度(weight)以產生分數