دانشگاه پیام نور
Payam Nour University

ارائه سمینار تحقیق و تتبع نظری

بسم الله الرحمن الرحیم

دانشگاه پیام نور
Payam Nour University

ارائه سمینار تحقیق و تتبع نظری

عنوان سمینار

بهبود نسبت فشرده سازی متن برای متن اسکی

(بررسی و مرور)

دانشگاه پیام نور
Payam Nour University

ارائه سمینار تحقیق و تتبع نظری

استاد راهنما

دکتر سیدعلی رضوی ابراهیمی

نگارنده

مریم سادات موردگر

شهریور ۱۴۰۰

فهرست

- تعریف مسئله و اهداف تحقیق
- بررسی مفاهیم فشرده سازی
- مروری بر کارهای انجام شده
- جمع بندی و پیشنهادات

تعریف مسئله و اهداف تحقیق

تعریف مسئله

انتقال حجم زیادی از داده ها از طریق اینترنت کار زمان بری است. فشرده سازی داده ها حجم فایل را کاهش می دهد، پس انتقال آن از طریق اینترنت سریعتر خواهد بود.

برخی از الگوریتم های فشرده سازی بدون اتلاف داده و برخی با اتلاف داده هستند. بطور مثال فشره سازی فایل ها با فرمت تصویر و ویدئو با اتلاف داده می باشد ولی برای فشرده سازی داده های متنی باید از الگوریتم های بدون اتلاف استفاده گردد.

اهداف تحقیق

در این تحقیق تلاش شده تا روشی مخصوص فشرده سازی متن ASCII ارائه شود. این روش از ترکیب سه الگوریتم فشرده سازی محبوب دنیا استفاده می کند تا با فشرده سازی بیشتر داده، نسبت فشرده سازی متن ASCII را بهبود دهد.

روش ارائه شده از ترکیب کدگذاری دیکشنری، فشرده سازی ASCII و کدگذاری هافمن استفاده می کند.

روش فشرده سازی ترکیبی TCS است که مخفف Tool-less Compression System می باشد.

بررسی مفاهیم فشرده سازی

انواع فشرده سازی داده

بدون اتلاف

دراین روش وقتی فایل از حالت فشرده خارج می شود، با فایل اولیه یکسان خواهد بود و هیچ داده ای از بین نمی رود.

- برای فایل های متنی مناسب است.
- کدگذاری دیکشنری، فشرده سازی متن ASCII و کدگذاری هافمن از این روش استفاده می کنند.

با اتلاف

دراین روش برخی داده ها هنگام فشرده سازی از بین می روند و هنگامی که فایل از حالت فشرده خارج می شود، داده های از دست رفته قابل بازیابی نیستند.

- برای فایل صوتی، تصویر و ویدئو مناسب می باشد.

تعریف

کدگذاری دیکشنری از رشته های تکراری متن استفاده می کند و آن ها را با منابع جایگزین می کند.

- الگوریتم های LZ77، LZ78، LZW و Re-pair از کدگذاری دیکشنری استفاده می کنند.

فشرده سازی ASCII یک کدگذاری متنی است که به طور گسترده برای متنی که کاراکترها و نمادهایی با تنوع زیاد ندارد، استفاده می شود.

- الگوریتم Shoco از فشرده سازی ASCII استفاده می کند و برای متن های مناسب خیلی سریع است.

کدگذاری هافمن کدگذاری هافمن یک رمزگذار آنتروپی است، به این معنی که داده ها را بر اساس تکرار نماد، فشرده می کند و یک درخت دودویی می سازد.

- کدگذاری هافمن اغلب با کدگذاری حسابی مقایسه می شود که کدگذاری هافمن سریع تر از کدگذاری حسابی است، اما کدگذاری حسابی به طور کلی نسبت فشرده سازی بهتری دارد.

مروری بر کارهای انجام شده

روش ها و متدولوژی ها

روش تحقیق

- این پروژه از روش تحقیق تجربی استفاده می کند، زیرا آزمایش بخش مهمی از پروژه است.

استراتژی تحقیق

- به همان دلیل استفاده از روش تجربی، استراتژی تحقیق نیز استراتژی تجربی است.

جمع آوری داده ها

- این پروژه از آزمایشات برای جمع آوری داده ها استفاده می کند.

تجزیه و تحلیل داده ها

- در این پروژه برای تجزیه و تحلیل داده ها از روش آمار استفاده شده است.

الزامات سیستم

الزامات عملکردی TCS

- در طول فشرده سازی هیچ داده ای از بین نمی رود.
- TCS می تواند متن را با رمزگذاری های معمول و غیرمعمول فشرده کند.
- ماژول های TCS به راحتی متصل می شوند.

الزامات غیرعملکردی TCS

- TCS می تواند فایل های بالای ۲۰ مگابایت را فشرده کند.
- TCS می تواند فایلها را حداقل به نصف اندازه اصلی خود فشرده کند.
- تمام کدهای شخص ثالثی که در TCS استفاده می شود باید منبع باز باشد.

مجموعه داده ها (Data set)

مجموعه داده های مورد استفاده در پایان نامه مورد بررسی، شامل ۶ متن با ویژگی های مختلف است:

۱- یک فایل XML ویکی پدیا. رمزگذاری UTF-8 ؛ ۲۱.۶ مگابایت

۲- یک فایل XML ویکی پدیا. کدگذاری cpo37 ؛ ۱۸.۱ مگابایت

۳- یک فایل کد نوشته شده با C. کدگذاری ASCII؛ ۶۴ کیلوبایت

۴- کتابی که به زبان انگلیسی نوشته شده است. رمزگذاری UTF-8 ؛ ۶۸۰ کیلوبایت

۵- کتابی که به زبان ایتالیایی نوشته شده است. رمزگذاری UTF-8 ؛ ۶۲۶ کیلوبایت

۶- کتابی که به زبان چینی نوشته شده است. رمزگذاری UTF-8 ؛ ۲۸۵ کیلوبایت

طراحی TCS

- ACM (فشرده سازی متن ASCII) فقط قادر به فشرده سازی کاراکترهای اسکی است. و خروجی رمزگذار دیکشنری یا ماژول های کدگذاری هافمن، داده های دودویی هستند که شامل کاراکترهای اسکی نیستند، بنابراین ابتدا ACM استفاده می شود.

- خروجی ACM یک فایل متنی در کدگذاری cpo37 است. این خروجی به کدگذار دیکشنری ارسال می شود که زیر رشته های تکراری را در متن جستجو می کند.

- سپس خروجی دودویی از رمزگذار دیکشنری به ماژول کدگذاری هافمن _که مقادیر بایت را که بیشتر از بقیه استفاده می کند_ ارسال می شود.

- فایل خروجی از ماژول کدگذاری هافمن آخرین فایل فشرده شده از TCS است.

پیاده سازی کدگذاری دیکشنری و انتخاب بهترین الگوریتم

برای پیاده سازی این روش از سه الگوریتم LZ77 (با پیاده سازی پایتون) و LZ77 (با پیاده سازی C) و LZW (با پیاده سازی C)، روی مجموعه داده منتخب استفاده شده است.

از مقایسه این سه الگوریتم نتایج زیر حاصل شد:

- اجرای پایتون از الگوریتم LZ77 بدترین نسبت فشرده سازی را دارد.
- پیاده سازی C برای الگوریتم LZ77 دارای نسبت فشرده سازی بسیار بهتری است.
- پیاده سازی LZW نسبت به پیاده سازی C در LZ77 برای برخی از متون دارای نسبت فشرده سازی بالاتری است، اما برای متون دیگر نسبت کمتری دارد.

بنابراین پیاده سازی C الگوریتم LZW به عنوان ماژول کدگذاری دیکشنری استفاده شده است.

پیاده سازی کدگذاری هافمن و انتخاب بهترین الگوریتم

برای کدگذاری هافمن از دو پیاده سازی C، که توسط کاربران گیت هاب Yaikhom و Richardson ساخته شده و یک پیاده سازی پایتون به نام Dahuffman که توسط کاربر گیت هاب Lippens ساخته شده است، استفاده شده است.

از مقایسه این سه الگوریتم نتایج زیر حاصل شد:

- دو پیاده سازی C نتایج دقیقا یکسانی را به دست آوردند.
- پیاده سازی پایتون، Dahuffman، نسبت فشرده سازی کمی بهتر از دو پیاده سازی C برای کتاب انگلیسی و ایتالیایی، اما نسبت بهتری برای کتاب چینی دارد.

Dahuffman ، دارای بالاترین نسبت فشرده سازی متوسط پیاده سازی ها است و بنابراین به عنوان ماژول کدگذاری هافمن برای TCS استفاده شده است.

نتایج حاصل از بررسی TCS

Data set	مجموعه داده ها	ACM	LZW	Huffman coding	Total ratio
XML file (21.6 MB)	فایل XML	1.19	1.6	1.01	**1.93**
cpo37 encoded XML file (18.1 MB)	فایل XML کدگذاری cpo37	1	1.6	1.01	**1.62**
C code file (64 KB)	فایل کد C	1.21	1.83	1.01	**2.23**
English book (680 KB)	کتاب انگلیسی	1.46	1.36	1	**1.99**
Italian book (626 KB)	کتاب ایتالیایی	1.29	1.57	1	**2.02**
Chinese book (285 KB)	کتاب چینی	1	1.59	1	**1.59**
Average	میانگین	1.19	1.59	1.01	**1.9**

نسبت فشرده سازی به دست آمده از ماژول ترکیبی TCS

نتایج حاصل از مقایسه TCS با برخی فشرده سازی های موفق

Data set	مجموعه داده ها	TCS	Bzip2	Zip	7-Zip	Gzip
XML file (21.6 MB)	فایل XML	1.93	3.79	2.96	4.24	2.96
cpo37 encoded XML file (18.1 MB)	فایل XML کدگذاری cpo37	1.62	3.12	2.45	3.48	2.45
C code file (64 KB)	فایل کد C	2.23	4.22	3.88	4.25	3.91
English book (680 KB)	کتاب انگلیسی	1.99	3.52	2.65	3.18	2.65
Italian book (626 KB)	کتاب ایتالیایی	2.02	3.54	2.66	3.14	2.66
Chinese book (285 KB)	کتاب چینی	1.59	2.59	2.01	2.35	2.01
Average	میانگین	1.9	3.46	2.77	3.44	2.77

از ارزیابی به دست آمده نتایج زیر حاصل شد:

- **TCS نسبت به سایر برنامه ها برای هر فایل در مجموعه داده نسبت فشرده سازی کمتری دارد.**
- **Bzip2 و 7-Zip بالاترین میانگین نسبت فشرده سازی را دارند زیرا از الگوریتم های متفاوتی نسبت به Zip و Gzip استفاده می کنند که بر اساس الگوریتم DEFLATE است.**

نتایج حاصل از مقایسه ACM+DEFLATE

سوالی که از ارزیابی قبل حاصل شد اینست که آیا استفاده از ACM به عنوان پیش پردازنده باعث افزایش نسبت فشرده سازی الگوریتم DEFLATE می شود یا خیر؟

Data set	مجموعه داده ها	Zip	ACM + Zip	Gzip	ACM + Gzip
XML file (21.6 MB)	فایل XML	2.96	2.92	2.96	2.92
C code file (64 KB)	فایل کد C	3.88	3.77	3.91	3.82
English book (680 KB)	کتاب انگلیسی	2.65	2.66	2.65	2.66
Italian book (626 KB)	کتاب ایتالیایی	2.66	2.58	2.66	2.58
Average	میانگین	3.04	2.98	3.05	3

ارزیابی انجام شده نشان می دهد که استفاده از ACM باعث افزایش نسبت فشرده سازی الگوریتم DEFLATE نخواهد شد.

جمع بندی و پیشنهادات

جمع بندی

نتایج حاصل از ارزیابی TCS نشان می دهد که در حالت فعلی، مدعی سایر برنامه های فشرده سازی عمومی نیست. ولی فشرده سازی متوسط بهتری نسبت به برخی مدل های فشرده سازی عمومی دارد. بنابراین ACM می تواند مدعی رشته تخصصی تر فشرده سازی ASCII باشد. ارزیابی ACM نشان می دهد که می توان آن را در هر شکلی از متن بدون افزایش اندازه فشرده استفاده کرد، اما نسبت فشرده سازی قابل توجهی فقط در متون سنگین ASCII به دست می آورد. با این وجود، نسبت فشرده سازی ACM در مقایسه با برنامه های فشرده سازی عمومی بسیار کوچک است. ترکیب ACM با برنامه های فشرده سازی مبتنی بر DEFLATE ممکن است نسبت فشرده سازی را افزایش دهد.

پیشنهادات و کارهای آینده

- کارهای آینده در TCS شامل یافتن یک ماژول کدگذاری هافمن است که بتواند خروجی دودویی را از ماژول کدگذاری دیکشنری فشرده کند.
- آزمایش الگوریتم هایی که توسط DEFLATE برای بهبود نسبت فشرده سازی استفاده نشده است، نیز کار آینده است.
- برای افزایش نسبت فشرده سازی می توان پیشرفت هایی در ACM انجام داد. ACM می تواند فشرده سازی کاراکترهای غیر ASCII را بهبود بخشد، زیرا تکنیک فعلی بهینه نیست.
- همچنین با بازنویسی کد در C می توان افزایش تأخیر ACM را بهبود داد.

با تشکر از توجه شما